



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ**  
**ΜΗΧΑΝΙΚΩΝ Η/Υ**

## **Διπλωματική Εργασία**

**Συνεργατική Προσέγγιση εύρεσης θέσεων στάθμευσης σε έξυπνες πόλεις:  
Αλγόριθμοι, σχεδίαση και υλοποίηση συστήματος υποστήριξης.**

**Cooperative approach for finding parking spaces in smart cities:  
Algorithms, design and backend implementation**

**Συγγραφείς**  
*Γεώργιος Μακαντάσης*

**Επιβλέποντες Καθηγητές**  
Κατσαρός Δημήτριος  
Τσουκαλάς Ελευθέριος

# Abstract

Σε αυτή την διπλωματική θα παρουσιάσουμε την λειτουργία μιας εφαρμογής σε περιβάλλον Android η οποία θα λειτουργεί ως ένας βοηθός parking στους χρήστες-οδηγούς.

Ο κάθε οδηγός θα μπορεί να βλέπει στην οθόνη του κινητού του τα κοντινότερα σε αυτόν σημεία στάθμευσης τα οποία θα δημοσιεύονται από άλλους χρήστες-πεζούς. Συνεπώς η εφαρμογή έχει συνεργατικό χαρακτήρα καθώς στηρίζεται στην άμεση αλληλεπίδραση των χρηστών μεταξύ τους.

**Keywords:** Smart Cities, Smart Parking, Android Application, Cooperative Approach,



# Περιεχόμενα

<b>1.Εισαγωγή.....</b>	<b>6</b>
1.1 Smart Cities.....	6
1.2 Smart Parking.....	8
1.3 Υπάρχουσες Λύσεις.....	8
1.4 Εναλλακτική Προσέγγιση.....	9
<b>2. Application.....</b>	<b>10</b>
2.1 Κινητές Συσκευές.....	10
2.2 Αρχιτεκτονική Android.....	11
2.3 Τεχνολογίες.....	12
2.3.1 Location.....	13
2.3.2 Map.....	14
2.3.3 Cloud Messaging.....	14
2.3.4 Geocoding.....	15
2.3.5. Server Technologies.....	16
2.4 User Interface.....	17
2.4.1 Login Activity.....	18
2.4.2 Select Action Activity.....	18
2.4.3 Driver Activity.....	19
2.4.4 Pedestrian Activity.....	20
2.4.5 Screenshots.....	21
<b>3.Σενάρια Λειτουργίας.....</b>	<b>24</b>
3.1 Pedestrian Activity.....	24
3.2 Driver Activity.....	27
3.3 Ειδικές Περιπτώσεις.....	29
<b>4.Λειτουργίες Server.....</b>	<b>31</b>
4.1 Πίνακες SQL.....	31
4.2 User Authentication.....	32
4.3 Διαχείριση Driver Mode.....	32
4.4 Διαχείριση Pedestrian Mode.....	33
4.5 Διαχείριση Βάσης Δεδομένων.....	33
4.5.1 Δέσμευση ή αποδέσμευση θέσης.....	33
4.5.2 Ψηφοφορία.....	34

4.6 Οργάνωση Ειδοποιήσεων.....	35
4.6.1 Προσθήκη νέου σημείου.....	35
4.6.2 Αρνητική Ψηφοφορία σε μια υπάρχουσα θέση στάθμευσης.....	36
4.7 Αξιολόγηση της αξιοπιστίας των χρηστών μέσω της μεθόδου Wilson.....	36
4.8 Χρονολόγηση Σημείων.....	38
<b>5.Evaluation.....</b>	<b>40</b>
5.1 Μεθοδολογία.....	40
5.2 Πειράματα.....	41
<b>6. Συμπεράσματα-Επεκτάσεις.....</b>	<b>43</b>
6.1 Συμπεράσματα.....	43
6.2 Επεκτάσεις.....	43
<b>7. Βιβλιογραφία.....</b>	<b>44</b>

# 1.ΕΙΣΑΓΩΓΗ

## 1.1 Smart Cities

Ο σύγχρονος, αστικοποιημένος τρόπος ζωής έχει δημιουργήσει προκλήσεις και προβλήματα στην καθημερινότητα των πολιτών, κυρίως στα μεγάλα αστικά κέντρα. Τέτοιες προκλήσεις είναι. Η οικονομική ύφεση και τα υψηλά επίπεδα ανεργίας, σε συνδυασμό με την πληθυσμιακή έκρηξη μας κάνουν να αναθεωρήσουμε τον τρόπο και τις μεθόδους διαχείρισης των αναγκών στέγασης και μετακίνησης. Η κλιματική αλλαγή κάνει επιβεβλημένη την βέλτιστη διαχείριση πόρων και την μείωση των εκπομπών ρύπων.

Η διαχείριση αυτών των προκλήσεων είναι συνυφασμένη με την καλή ποιότητα ζωής των κατοίκων και την βέλτιστη διαχείριση των πόρων και του ρυθμού ανάπτυξης των οικονομιών καθώς μεγάλο ποσοστό του ΑΕΠ παράγεται σε μεγάλα αστικά κέντρα (Αγγίζει μέχρι το 80% στο Ηνωμένο Βασίλειο).

Βασικό στοιχείο μιας αποδοτικής αντιμετώπισης των προκλήσεων είναι η εκμετάλλευση των νέων τεχνολογικών επιτευγμάτων έχοντας σαν επίκεντρο την πληροφορία, η οποία γίνεται όλο και πιο αξιόπιστη και μαζικά προσβάσιμη από το σύνολο του πληθυσμού.

Ένα περιβάλλον το οποίο διαχειρίζεται την δημόσια πληροφορία, με σκοπό την παροχή βελτιστοποιημένων υπηρεσιών στον τομέα της εξοικονόμησης ενέργειας και μετακίνησης, μπορεί να χαρακτηριστεί ως έξυπνη πόλη (Smart City) .

Βασικό χαρακτηριστικό και ταυτόχρονα επιδίωξη μιας έξυπνης πόλης είναι η ενεργός και συνεχής συμμετοχή όσο το δυνατόν μεγαλύτερης μερίδας του πληθυσμού στην παροχή πληροφορίας. Όσο μεγαλύτερη πληροφορία εισάγεται στο σύστημα αυτό γίνεται ολοένα και πιο αξιόπιστο με υψηλότερου επιπέδου υπηρεσίες.

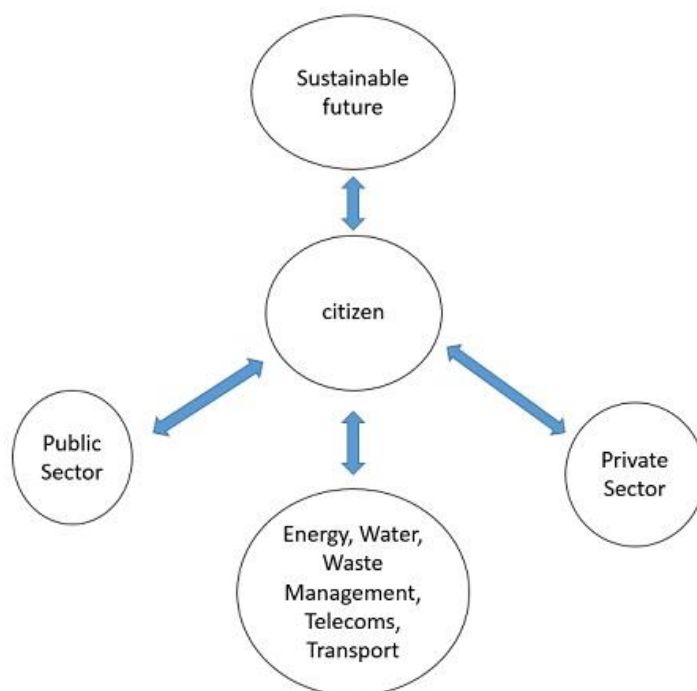


Figure 1: Smart City Diagram

Για να λειτουργήσει σωστά το μοντέλο της έξυπνης πόλης προϋποθέτει την εκμετάλλευση της πληροφορίας από την υπάρχουσα τεχνολογική υποδομή, η οποία περιέχει τα εξής χαρακτηριστικά:

- Σύγχρονη Ψηφιακή υποδομή για γρήγορη και ασφαλή προσπέλαση δεδομένων από τους πολίτες (Smart Phones, PC, Cellular Networks)
- Η παραδοχή ότι οι υπηρεσίες που προσφέρονται έχουν σαν επίκεντρο τον πολίτη έτσι ώστε να απολαμβάνουν όλοι υψηλής ποιότητας υπηρεσίες. Ένα τέτοιο παράδειγμα είναι η επέκταση των υπηρεσιών παροχής δικτύωσης και στις πιο απομακρυσμένες περιοχές.
- Η φυσική υποδομή σε μια πόλη όπως τα δίκτυα αισθητήρων (πχ Smart Grid, VANETS) επιτρέπουν στους παρόχους υπηρεσιών να έχουν μια συνεχή παρατήρηση των δεδομένων παρέχοντας τις πληροφορίες όπου χρειάζεται.
- Συνεχή εξέλιξη των τεχνολογικών συστημάτων που χρησιμοποιούνται και βασίζονται σε παρατηρήσεις από τρίτους και στον πειραματισμό.
- Δημόσια Παρουσίαση των στατιστικών και των αποτελεσμάτων στους πολίτες, πράγμα που μπορεί να λειτουργήσει σαν κίνητρο για την συνέχιση και εξέλιξη του μοντέλου της έξυπνης πόλης.



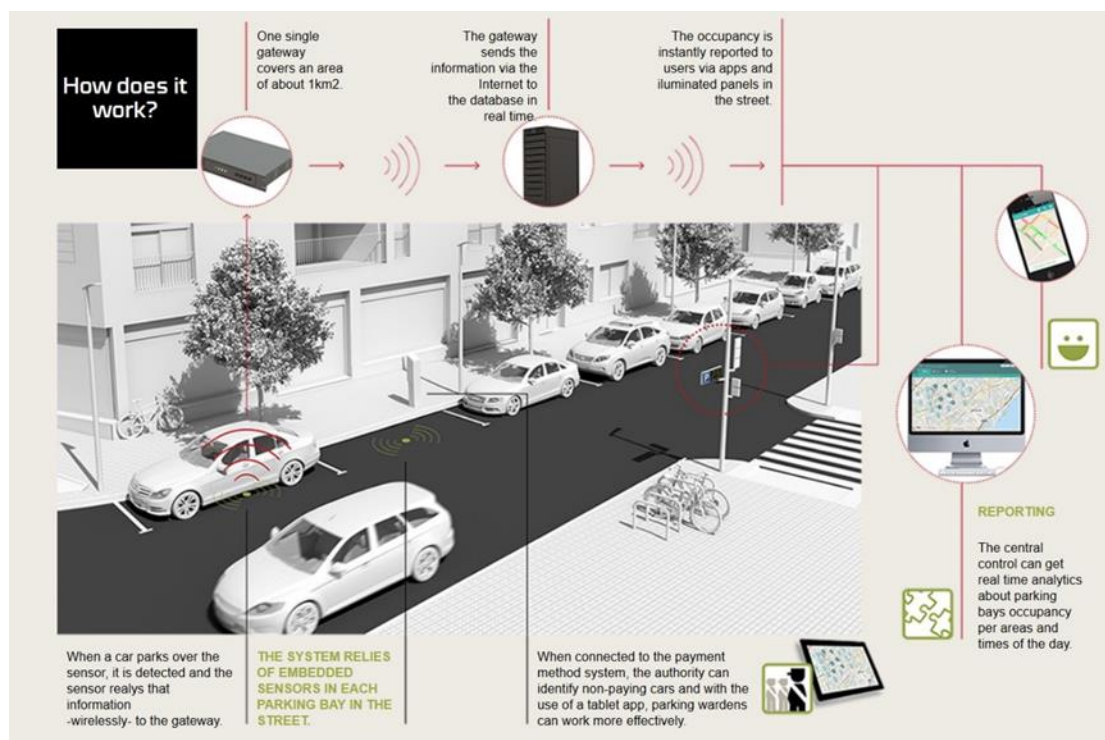
## **1.2 Smart Parking**

Οι έντονοι ρυθμοί ζωής στις σύγχρονες πόλεις δημιουργούν επιπλέον προκλήσεις και προβλήματα στην καθημερινότητα των πολιτών. Ένα από αυτά τα προβλήματα είναι η δυσκολία στην γρήγορη εύρεση θέσης parking που επηρεάζει αρνητικά τόσο τον οδηγό (άγχος και stress) όσο και το περιβάλλον καθώς οι περιττές μετακινήσεις κατά τη διάρκεια της αναζήτησης στάθμευσης αυξάνουν σε μεγάλο βαθμό τις εκπομπές ρύπων, υποβαθμίζοντας την ποιότητα ζωής των πολιτών. Επιπλέον μια ακόμη συνέπεια των περιττών μετακινήσεων είναι η δημιουργία κυκλοφοριακής συμφόρησης στα αστικά κέντρα, που με τη σειρά της επιβραδύνει τις οικονομικές δραστηριότητες. Συνεπώς, η αντιμετώπιση του παραπάνω προβλήματος σε ένα περιβάλλον έξυπνης πόλης με τον πολίτη στο επίκεντρο είναι επιβεβλημένη.

## **1.3 Υπάρχουσες λύσεις**

Σε αρκετές σύγχρονες, οικονομικά ανεπτυγμένες πόλεις το πρόβλημα της αποδοτικής διαχείρισης των θέσεων παρκινγκ λύνεται. Έχουν τοποθετηθεί αισθητήρες κατά μήκος όλων των δρόμων της περιοχής που εφαρμόζεται το σύστημα. Μέσω αυτών των αισθητήρων χαρτογραφούνται οι θέσεις παρκινγκ και παρέχουν την πληροφορία στο σύστημα, το οποίο με τη σειρά του, μέσω cloud υπηρεσιών την μεταβιβάζει στους οδηγούς που επιθυμούν να χρησιμοποιήσουν την υπηρεσία. Η τεχνολογία αυτή είναι βέλτιστη καθώς παρέχει τα δεδομένα αξιόπιστα και σε πραγματικό χρόνο.





### 1.4 Εναλλακτική προσέγγιση

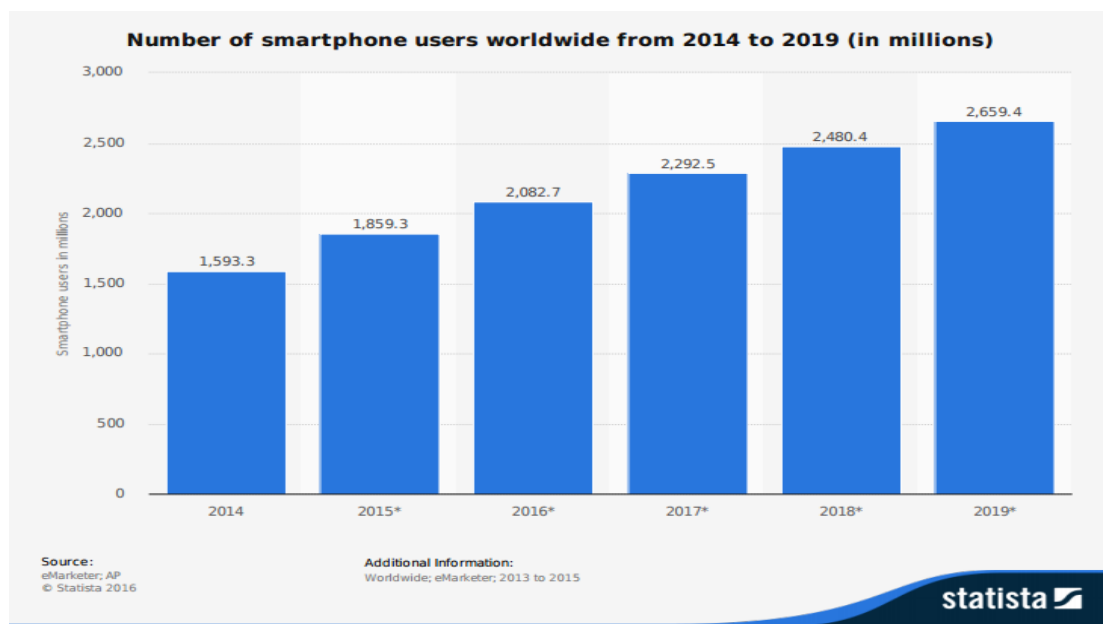
Η παραπάνω λύση πιθανότατα δεν μπορεί να εφαρμοστεί προς το παρόν, παρά μόνο σε πολύ περιορισμένες περιοχές και συνοικίες, λόγω του πολύ υψηλού κόστους. Στα πλαίσια αυτής της διπλωματικής εργασίας προτείνουμε ένα διαφορετικό τρόπο προσέγγισης του προβλήματος, το οποίο έχει σαν κεντρικό πάροχο πληροφοριών, αντί για ένα εκτενές δίκτυο αισθητήρων, τους κοινωνικά ευαισθητοποιημένους πολίτες, οι οποίοι άλλωστε αποτελούν βασική προϋπόθεση για τη σωστή λειτουργία ενός περιβάλλοντος “smart city”.

Η λύση που προτείνουμε έχει συνεργατικό χαρακτήρα (cooperative approach) καθώς βασίζεται στους χρήστες για την εισαγωγή έγκυρων πληροφοριών, που αφορούν τις διαθέσιμες θέσεις στάθμευσης. Αυτές θα αποτελούν δημόσιες και ανώνυμες πληροφορίες οι οποίες θα είναι διαθέσιμες σε όσους οδηγούς τις χρειάζονται. Ο κάθε πολίτης, χρησιμοποιώντας το κινητό του τηλέφωνο, ενημερώνει για νέες ελεύθερες θέσεις στάθμευσης το σύστημα, το οποίο με τη σειρά του τις γνωστοποιεί στους κοντινότερους οδηγούς. Μέσω της συγκεκριμένης προσέγγισης, ο κάθε χρήστης με ανεπτυγμένη κοινωνική συνείδηση, καταβάλλοντας ελάχιστη προσπάθεια, συμβάλλει σημαντικά στην βελτίωση της ποιότητας ζωής του ίδιου αλλά και των συνανθρώπων του.

## 2. Application

### 2.1 Κινητές Συσκευές

Οι κινητές συσκευές είναι πλέον ευρέως διαδεδομένες στο κοινό με βασικά πλεονεκτήματα την εύκολη πρόσβαση στο διαδίκτυο, την υψηλή υπολογιστική ισχύ και το χαμηλό κόστος αγοράς. Τα προγραμματιστικά εργαλεία που παρέχουν κάνουν πιο εύκολη την προσθήκη νέων λειτουργιών.



*Figure 2: Πρόβλεψη χρήσης έξυπνων συσκευών έως το 2019*

Για την υλοποίηση της συνεργατικής προσέγγισης για smart parking επιλέξαμε το λειτουργικό σύστημα Android καθώς βάση στατιστικών είναι το πιο διαδεδομένο λειτουργικό σύστημα.

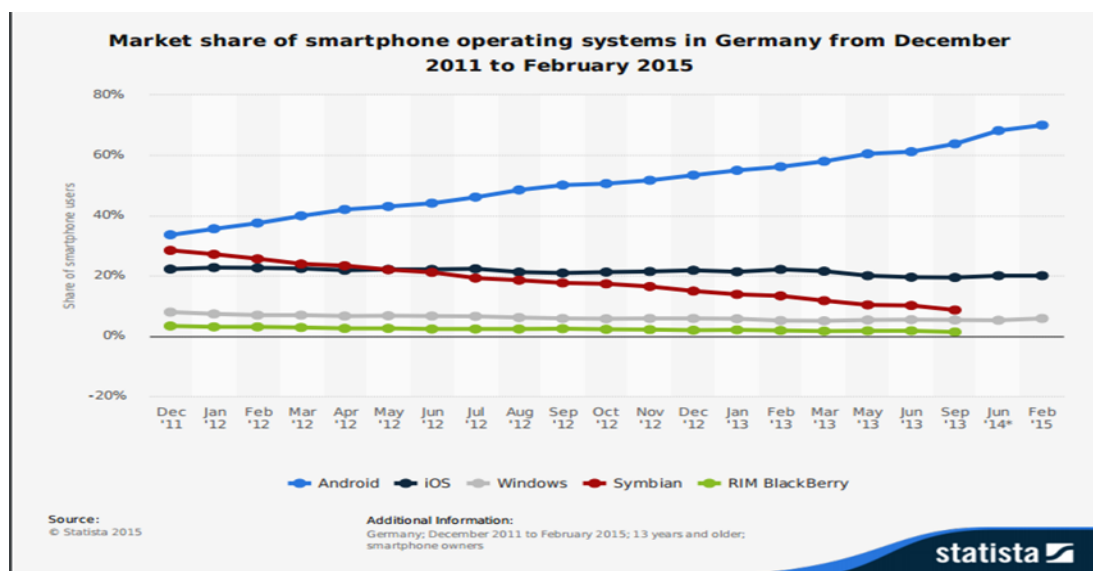


Figure 3: Μερίδιο αγοράς κινητών τηλεφώνων

## 2.2 Αρχιτεκτονική Android

Η αρχιτεκτονική του Λειτουργικού συστήματος Android OS χωρίζεται σε διακριτά επίπεδα

- Linux Kernel

Σε αυτό το επίπεδο γίνεται η αλληλεπίδραση με το hardware και η επικοινωνία με τις συσκευές μέσω των drivers.

- Μέσο Επίπεδο

### 1. Βιβλιοθήκες

Σε αυτό το τμήμα είναι εγκατεστημένες όλες οι απαραίτητες βιβλιοθήκες που χρησιμοποιεί το λειτουργικό (SQLite, OpenGL , Webkit κτλ).

### 2. Android Runtime

Σε αυτό το σημείο βρίσκεται το Java Virtual Machine.

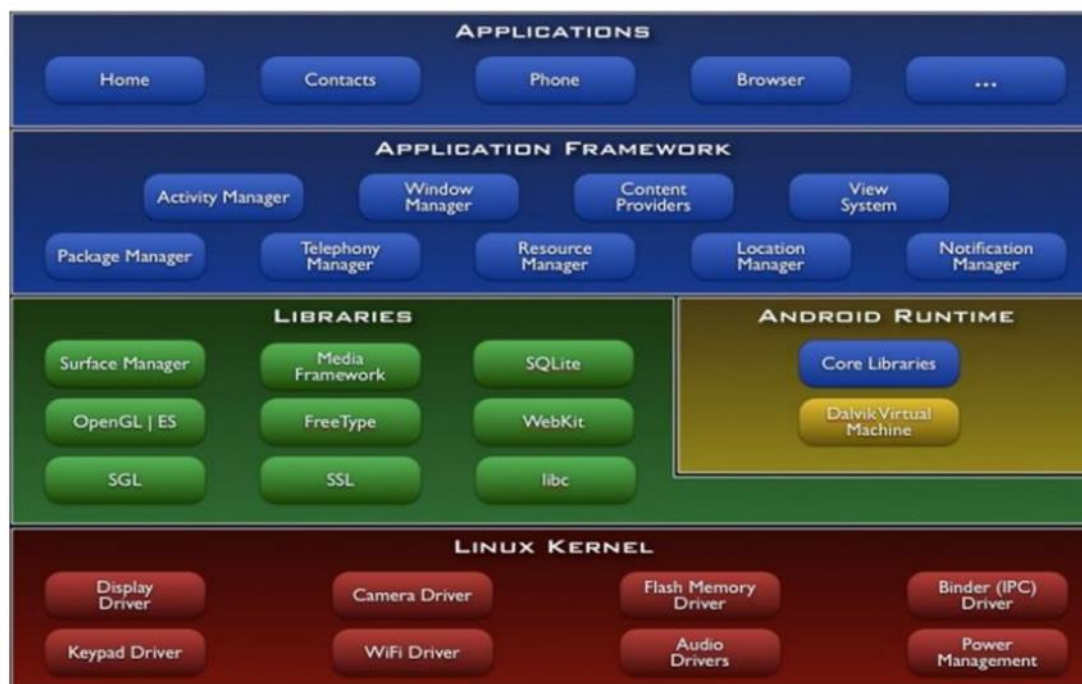
- Επίπεδο Εφαρμογής

### 1. Application Framework

Εδώ βρίσκονται τα βασικά στοιχεία αλληλεπίδρασης με την εφαρμογή μας. (Activity Manager, Location Manager, Resource Manager)

### 2. Applications

Σε αυτό το σημείο ανήκουν όλες οι βασικές εφαρμογές και ό,τι δημιουργούν τρίτοι.



### 2.3 Τεχνολογίες

Η Εφαρμογή χωρίζεται σε δυο βασικές λειτουργίες οι οποίες περιγράφονται σχηματικά



Figure 4: Λειτουργία Pedestrian

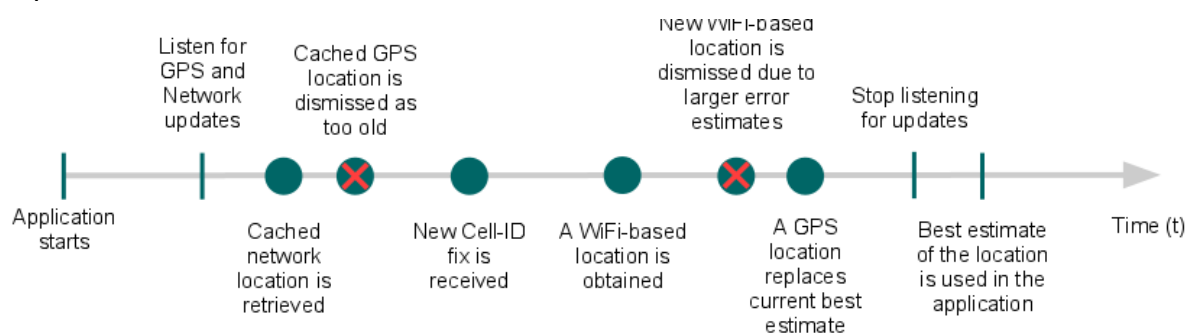


Figure 5: Λειτουργία Driver

### 2.3.1 Location

Το πρώτο βασικό βήμα είναι να λάβει η συσκευή γνώση της τοποθεσίας του χρήστη με σκοπό την όσο πιο ακριβή τοποθέτηση του σημείου στάθμευσης, ακολουθώντας την εξής διαδικασία:

Η ανάκτηση της τοποθεσίας του χρήστη κάποιες φορές είναι απαιτητική διαδικασία καθώς μπορεί να είναι ανακριβής ή να αποκλίνει από την πραγματική θέση. Συνεπώς είναι επιθυμητό να λαμβάνουμε τη θέση του χρήστη από όσο το δυνατόν περισσότερες πηγές (GPS, Network Location Service, WIFI Location Service). Η διαδικασία που ακολουθούμε για να εκτιμήσουμε με όσο το δυνατόν λιγότερο σφάλμα την τοποθεσία του χρήστη αποτυπώνεται σχηματικά παρακάτω:



*Figure 6: Διαδικασία Ανάκτησης τοποθεσίας με χρονική σειρά*

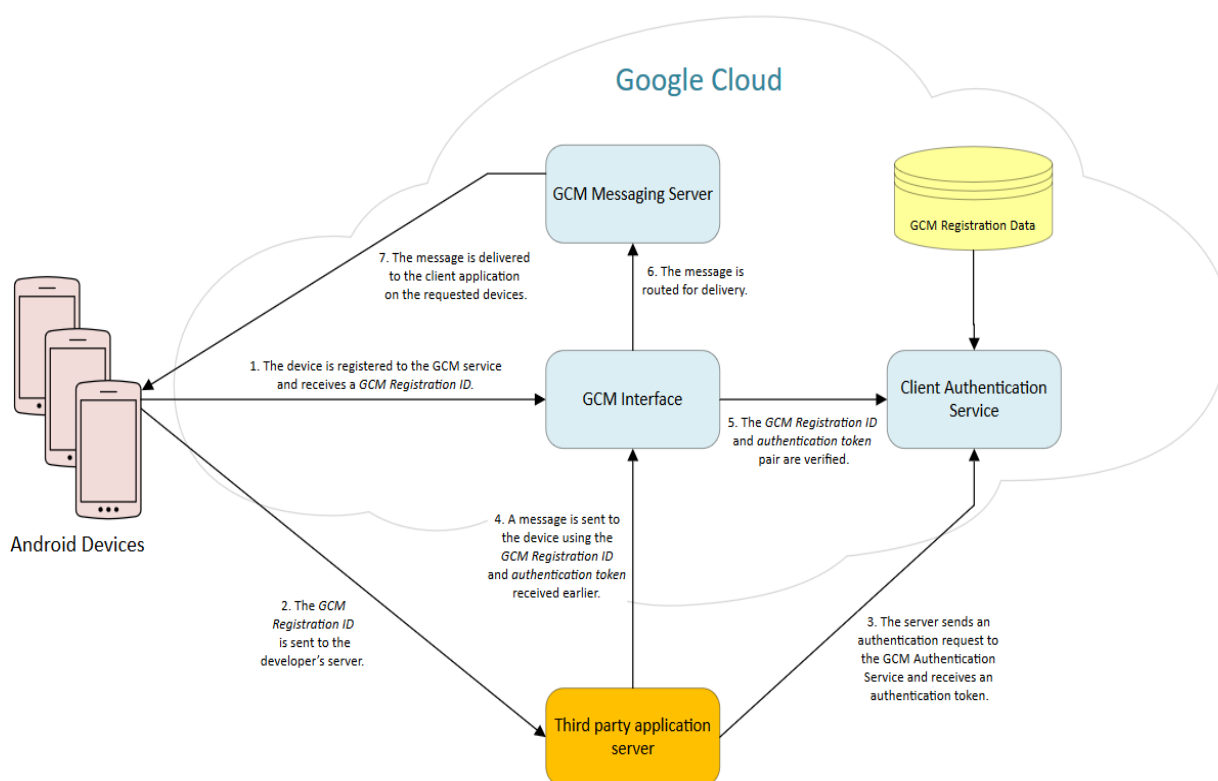
Με την εκκίνηση της εφαρμογής ενεργοποιούνται οι αισθητήρες gps και αναμένουμε τις συντεταγμένες τοποθεσίας. Μέχρι να γίνει αυτό η συσκευή ανακτά την τοποθεσία βάση του δικτύου κινητής τηλεφωνίας και από δίκτυο Wifi αν είναι διαθέσιμο. Σε επόμενο χρόνο έρχονται οι πληροφορίες μέσω GPS, οι οποίες αντικαθιστούν τις προηγούμενες εκτιμήσεις λόγω βέλτιστης ακρίβειας. Είναι σημαντικό επίσης να προσδιοριστεί η ιδανική συχνότητα λήψεως ενημερώσεων τοποθεσίας, έτσι ώστε και να ικανοποιούμε την ανάγκη για ακρίβεια αλλά και να ελαχιστοποιούμε την κατανάλωση ενέργειας.

### 2.3.2 Map

Για να οπτικοποιήσουμε με έναν φιλικό προς το χρήστη τρόπο τα γεωγραφικά δεδομένα που συλλέγουμε μέσω της λειτουργίας Location, χρησιμοποιούμε το Google Maps API το οποίο μας επιτρέπει να αναπαραστήσουμε ένα γεωγραφικό στίγμα σε μορφή Marker. Επίσης χρησιμοποιούμε επιπλέον λειτουργίες του API όπως για παράδειγμα υπολογισμός αποστάσεων μεταξύ δυο σημείων.



### 2.3.3 Google Cloud Messaging



Η συγκεκριμένη υπηρεσία μας επιτρέπει να στείλουμε εξατομικευμένα μηνύματα μέσω ειδοποιήσεων σε όσες συσκευές/clients επιθυμούμε.

Αρχικά η κάθε συσκευή μόλις ανοίξει την εφαρμογή εκτελεί την λειτουργία “Registration” της υπηρεσίας, μέσω του “GCM Interface” και λαμβάνει ένα μοναδικό ID επικοινωνίας. Στη συνέχεια το κινητό ενημερώνει τον Server της εφαρμογής μας με το ID που έλαβε και ολοκληρώνει το στάδιο της ταυτοποίησης. Σε περίπτωση που ο Server μας θέλει να επικοινωνήσει με κάποιον συγκεκριμένο χρήστη-client, αφού λάβει πρώτα έγκριση από το GCM authentication service της Google, τότε προωθεί το μήνυμα στο “GCM Interface” μαζί με το κλειδί επικοινωνίας του παραλήπτη και με την έγκριση που μόλις έλαβε. Τέλος, το μήνυμα αποστέλλεται στον παραλήπτη μέσω του “GCM Messaging Server”.

### 2.3.4 Geocoding

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "6",
          "short_name" : "6",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Βενιζέλου",
          "short_name" : "ΕΟ34Α",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Βόλος",
```

Figure 7: Τμήμα πακέτου Json

Json φαίνεται στην διπλανή εικόνα.

Η συγκεκριμένη υπηρεσία μας δίνει την δυνατότητα να μεταφράσουμε ένα γεωγραφικό στίγμα στην ακριβή του διεύθυνση πάνω στο χάρτη.

Για παράδειγμα το σημείο με συντεταγμένες (39.360785, 22.945561) μεταφράζεται σε ένα αρχείο Json με πολλές πληροφορίες. Από αυτό το αρχείο με κατάλληλους χειρισμούς ανακτούμε την οδό και τον αριθμό. Ένα μικρό ενδεικτικό τμήμα του αρχείου

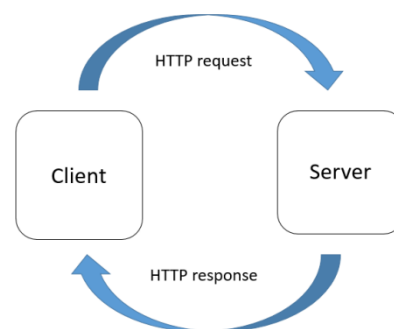


### **2.3.5 Server Technologies**

Έχουμε επιλέξει όσο το δυνατόν περισσότερες ενέργειες να γίνονται σε επίπεδο server παρά στο smartphone του χρήστη για λόγους εξοικονόμησης ενέργειας.

Η Τεχνολογία του Server είναι Tomcat v7 και οι λειτουργίες έχουν προγραμματιστεί με χρήση Java servlets μέσω του εργαλείου Eclipse. Ο Server τρέχει σε λειτουργικό σύστημα Linux Ubuntu εγκατεστημένο σε Virtual Machine του κεντρικού Server των εργαστηρίων του Τμήματος ΗΜΜΥ του Πανεπιστημίου Θεσσαλίας.

Ο Server είναι επιφορτισμένος με καθήκοντα διαχείρισης βάσης δεδομένων (MySQL), Διαχείρισης Sessions (Login/Logout). Η επικοινωνία του Server με τον client έχει request-response χαρακτήρα και όλες οι συνδέσεις γίνονται με χρήση του πρωτοκόλλου επιπέδου εφαρμογής http. Οι απαντήσεις του server προς τον client (Http Response) ακολουθούν την δομή αντικειμένων Json.





## 2.4 User Interface

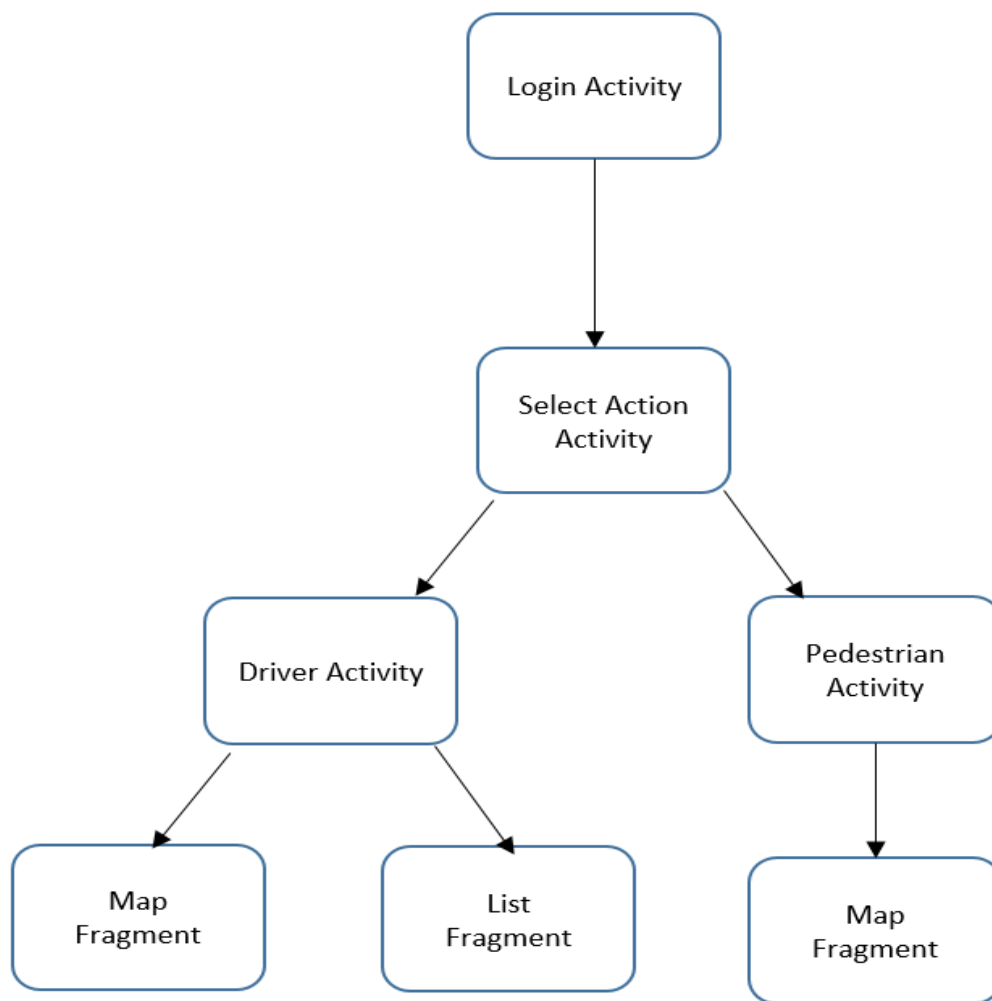


Figure 8: Δεντρικό διάγραμμα διεπαφής χρήστη

Για την υλοποίηση της βασικής λογικής η Mobile Εφαρμογής βασίζεται στους μηχανισμούς (Activities) που παρέχει το λειτουργικό σύστημα, με διεπαφή χρήστη δομημένη σε πολλαπλά τμήματα (Fragments), καθένα από τα οποία υλοποιεί μια ξεχωριστή λογική στην οθόνη η οποία ενσωματώνεται στο αντίστοιχο μηχανισμό (Activity). Έτσι έχουμε τις παρακάτω κύριες λειτουργίες:

### **2.4.1 Login Activity:**

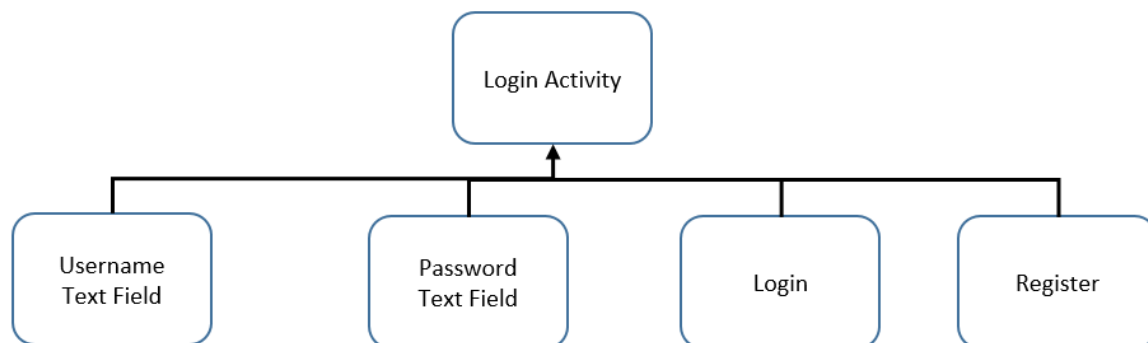


Figure 9: Διάγραμμα UI Login Activity

Ταυτοποίηση χρήστη και συνέχεια στα επόμενα βήματα.

### **2.4.2 Select Action Activity:**

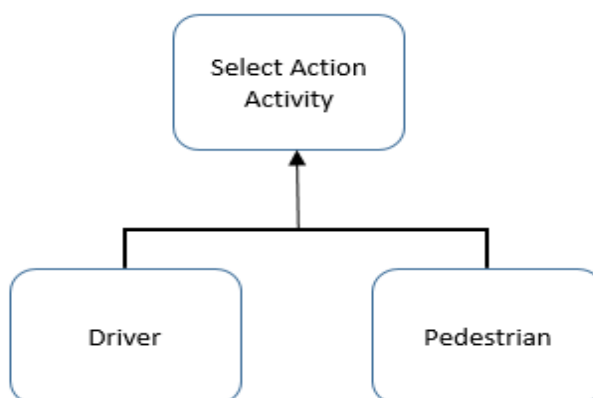


Figure 10: Διάγραμμα UI Selection Activity

Σε αυτό το στάδιο ο χρήστης επιλέγει την λειτουργία που θα επιτελέσει η εφαρμογή. Είτε θα μπει σε λειτουργία «οδηγού» ψάχνοντας θέση στάθμευσης ή θα επιλέξει ως «πεζός» να ενημερώσει το σύστημα για μια διαθέσιμη θέση στάθμευσης.

### 2.4.3 Driver Activity:

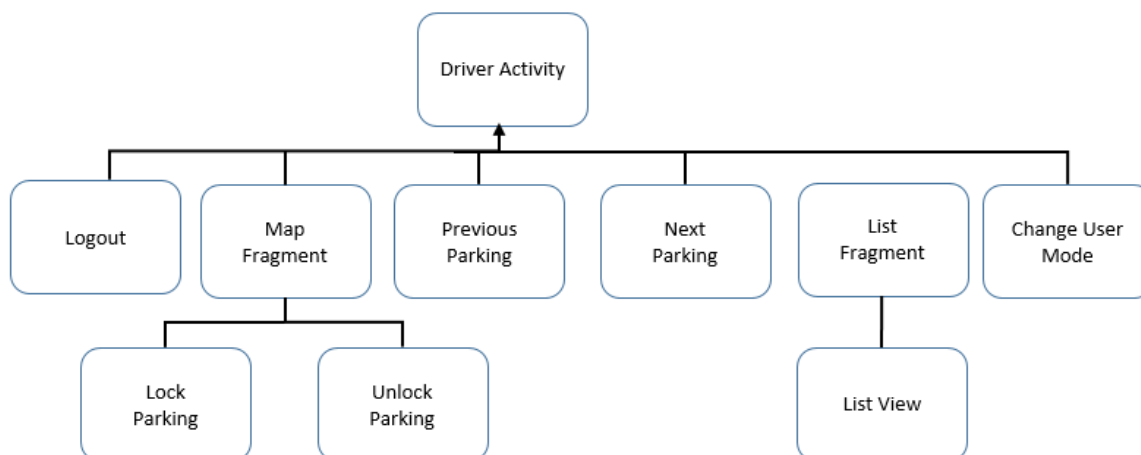


Figure 11: Διάγραμμα UI Driver Activity

Χωρίζεται σε δυο Fragments:

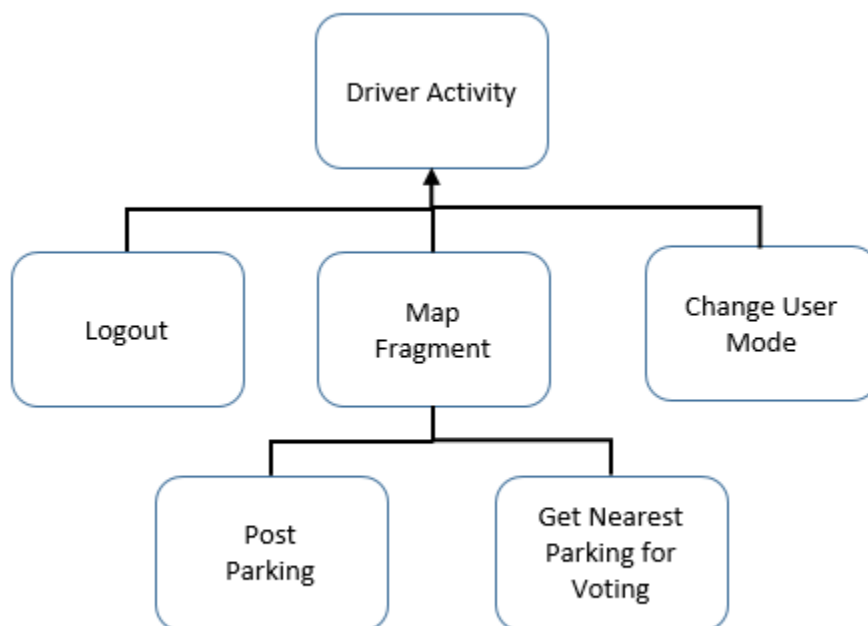
1. *Map Fragment:*

Υλοποιεί τη βασική διεπαφή του χρήστη με το αστικό περιβάλλον μέσω ενός χάρτη ο οποίος περιέχει όλες τις διαθέσιμες θέσεις στάθμευσης στην ευρύτερη περιοχή, παρέχοντας τις σχετικές πληροφορίες και όλες τις λειτουργίες δέσμευσης και αποδέσμευσης μιας υποψήφιας θέσης στάθμευσης.

2. *List Fragment*

Υλοποιεί έναν διαφορετικό τρόπο απεικόνισης των θέσεων στάθμευσης με μορφή λίστας και δίνει επιπρόσθετη πληροφορία σχετικά με την απόσταση κάθε θέσης από την τρέχουσα σε θέση του οδηγού.

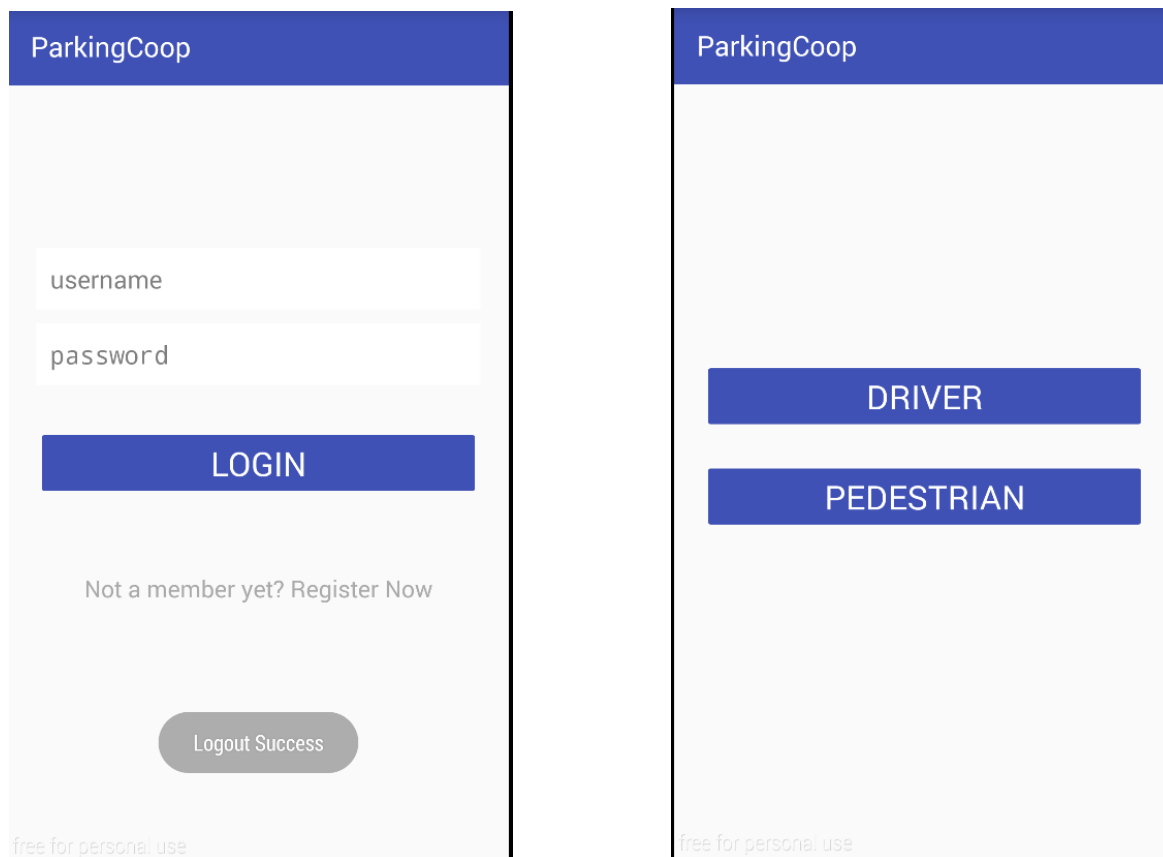
#### **2.4.4 Pedestrian Activity:**



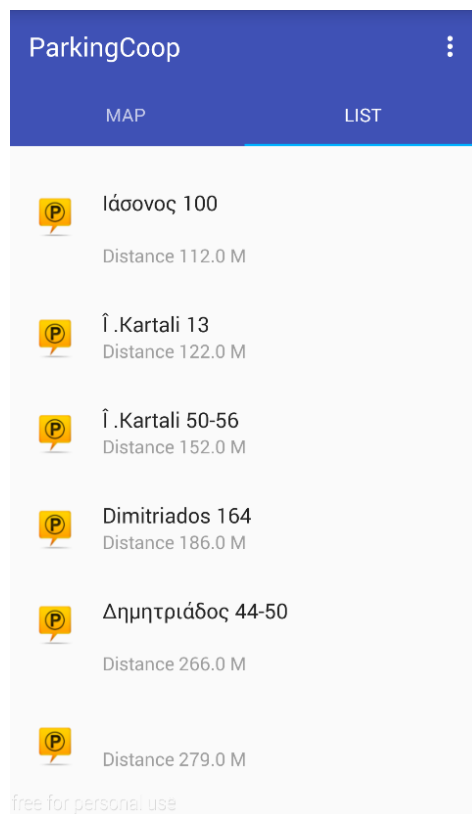
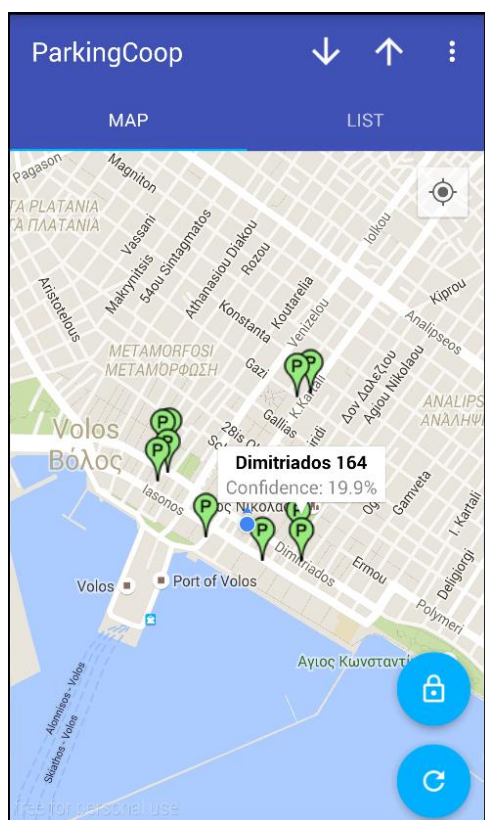
*Figure 12: Διάγραμμα UI Pedestrian Activity*

Υλοποιεί την βασική διεπαφή που απευθύνεται σε έναν πεζό οποίος είναι πρόθυμος να αξιολογήσει μια κοντινή θέση στάθμευση, να ενημερώσει για μια διαθέσιμη θέση τοποθετώντας την πάνω στον χάρτη κάνοντας χρήση του Map Fragment.

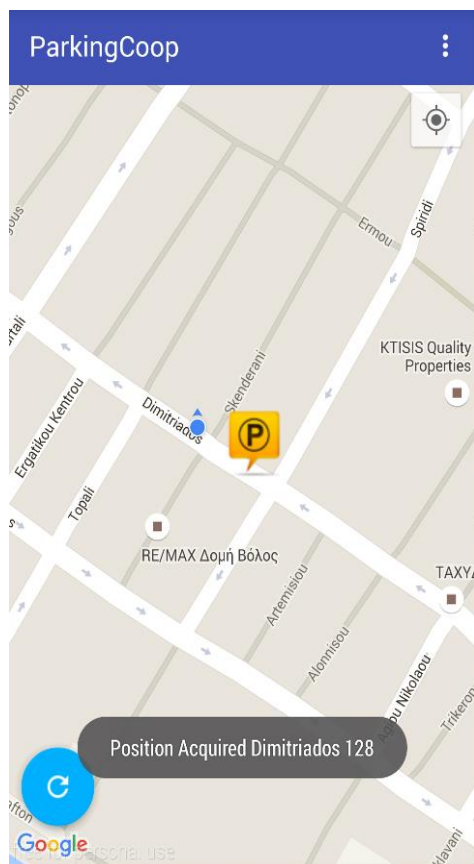
### 2.4.5 Screenshots



Αρχικά ο χρήστης εκτελεί την λειτουργία Login και η εφαρμογή μεταβαίνει στο Select Action Activity. Στη συνέχεια ο χρήστης επιλέγοντας το κατάλληλο κουμπί θα εκτελέσει την αντίστοιχη λειτουργία-Activity.



Στις παραπάνω εικόνες βλέπουμε πως δομείται η διεπαφή χρήστη ο οποίος εκτελεί λειτουργία Driver και μπορεί είτε να επιλέξει την διεπαφή Χάρτη (Map Fragment) είτε την διεπαφή Λίστας (List Fragment). Οι πράσινοι δείκτες αντιπροσωπεύουν τα κοντινότερα σημεία στάθμευσης, αντίστοιχα δεξιά δομούνται σε λίστα. Είτε μέσω της λίστας είτε μέσω του χάρτη μπορεί να γίνει δέσμευση της θέσης μέσω του κουμπιού Lock.



Τέλος η διεπαφή χρήστη ο οποίος εκτελεί λειτουργία δημοσίευσης θέσης στάθμευσης. Ο πορτοκαλί δείκτης είναι το σημείο που έχει επιλεγεί από τον χρήστη ως σημείο ελεύθερο προς στάθμευση.

### 3.ΣΕΝΑΡΙΑ ΛΕΙΤΟΥΡΓΙΑΣ

#### 3.1 Pedestrian Activity

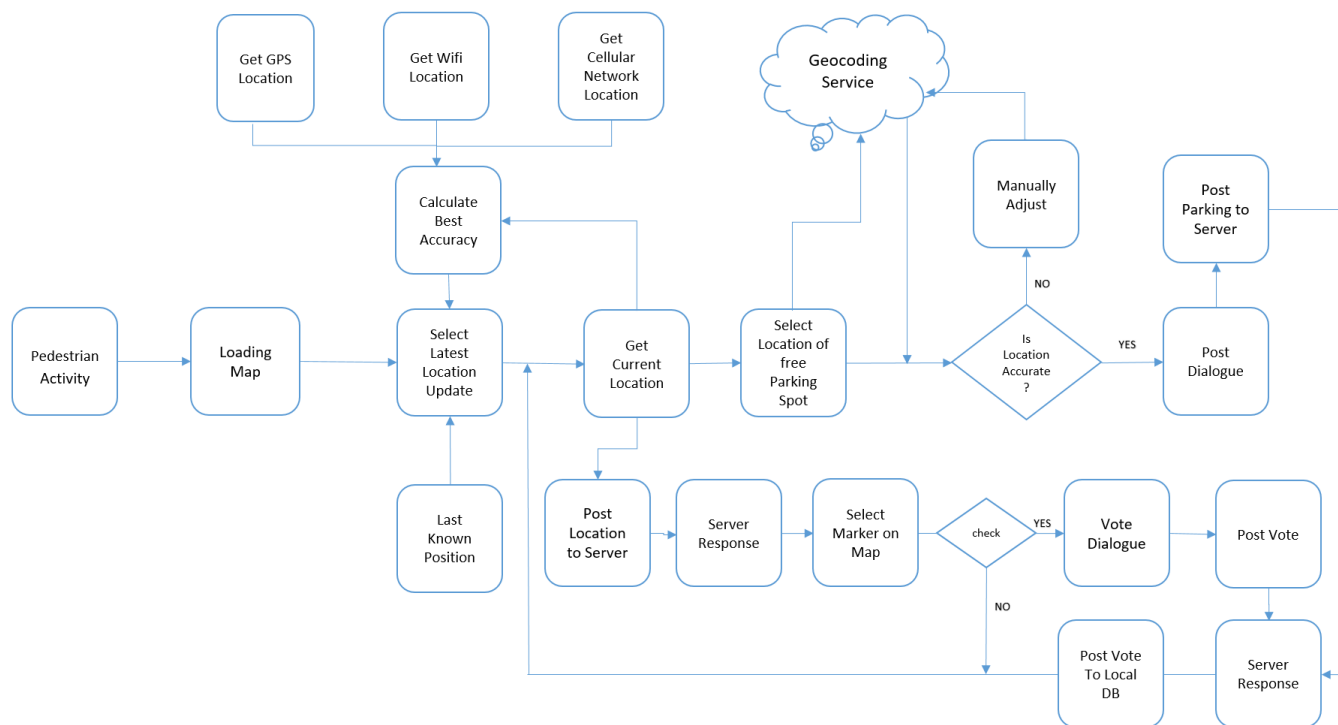


Figure 13: Διάγραμμα ροής Pedestrian Activity

Με την εκκίνηση της λειτουργία pedestrian η εφαρμογή αρχικοποιεί το Map Fragment και φορτώνει τον χάρτη από την υπηρεσία Google Maps. Στη συνέχεια ανακτάται η τοποθεσία του χρήστη με την εξής διαδικασία. Αρχικά ορίζεται σαν τρέχουσα τοποθεσία η τελευταία γνωστή τοποθεσία από την βάση δεδομένων της κινητής συσκευής και αναμένεται ανανέωση της παρούσας θέσης από τους αισθητήρες GPS,WIFI,ή κινητής τηλεφωνίας. Η συγκεκριμένη τοποθεσία επανεκτιμάται ανά τακτά χρονικά διαστήματα. Εφόσον γίνει εκτίμηση της τρέχουσας θέσης του χρήστη η εφαρμογή είναι έτοιμη να εκτελέσει τις εξής λειτουργικότητες:



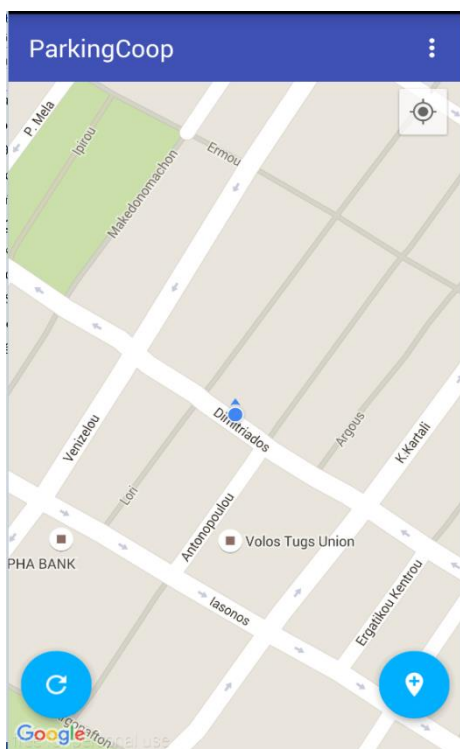


Figure 14: Αρχική οθόνη Pedestrian και εμφάνιση βασικών επιλογών

### 1. Ενημέρωση του συστήματος για μια διαθέσιμη θέση στάθμευσης.

Μόλις ο χρήστης πατήσει το αντίστοιχο κουμπί της διεπαφής τότε ανακτούμε το γεωγραφικό μήκος και γεωγραφικό πλάτος (Latitude, Longitude) το οποίο έχει εκτιμηθεί με όσο το δυνατόν βέλτιστη ακρίβεια στο προηγούμενο βήμα και εμφανίζεται ένα Marker πάνω στο χάρτη στη συγκεκριμένη τοποθεσία.

Για να γίνουν αυτά τα δεδομένα πιο φιλικά προς τη χρήση η εφαρμογή εκτελεί ασύγχρονα την ενέργεια Http Get request προς την υπηρεσία Geocoding δίνοντας παραμέτρους το αντίστοιχο γεωγραφικό μήκος και πλάτος. Η συγκεκριμένη υπηρεσία θα μετατρέψει τις γεωγραφικές συντεταγμένες σε Οδό και αριθμό και θα επιστρέψει τα δεδομένα σε αντικείμενο της μορφής Json.

Με δεδομένο ότι η εκτίμηση της τοποθεσίας του χρήστη σε ένα πυκνοκατοικημένο αστικό περιβάλλον κάποιες φορές εισάγει ανακριβή αποτελέσματα, ο χρήστης καλείται να εξακριβώσει την ακρίβεια της τοποθεσίας της θέσης στάθμευσης. Σε περίπτωση που η

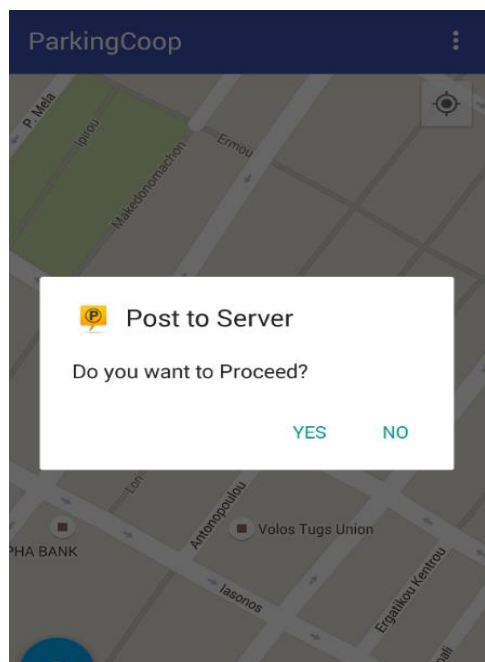


Figure 15: Διάλογος Επιβεβαίωσης Θέσης στάθμευσης.

θέση έχει εισαχθεί ανακριβώς πάνω στο χάρτη τότε η διεπαφή χρήστη του επιτρέπει να προβεί σε διορθωτικές κινήσεις μετακινώντας τον Marker σε έγκυρο σημείο. Η διαδικασία ανάκτησης της νέας διεύθυνσης από το τροποποιημένο στίγμα επαναλαμβάνεται. Στο σημείο αυτό ο χρήστης έχει προσδιορίσει τη θέση στάθμευσης και ενδεχομένως την έχει διορθώσει λόγω ανακρίβειας των γεωγραφικών συντεταγμένων. Στη συνέχεια το σύστημα ενημερώνεται για τη νέα θέση στάθμευσης μέσω ασύγχρονου http post request προς το Server δίνοντας παραμέτρους τις συντεταγμένες και την διεύθυνση που έχει ανακτηθεί από την υπηρεσία Geocoding. Εφόσον η

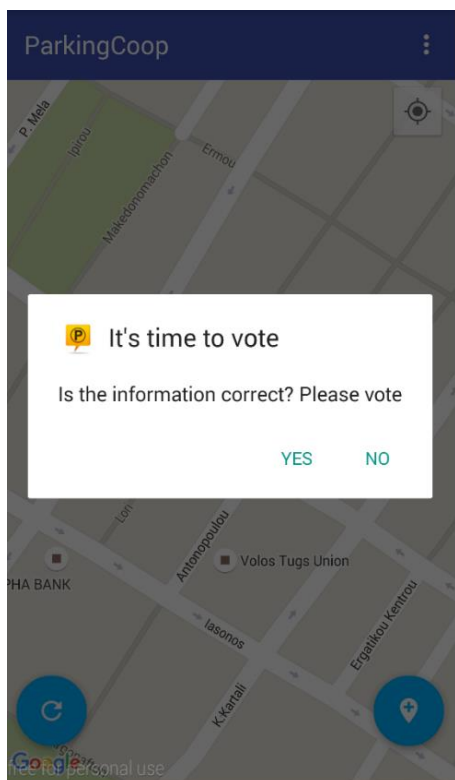


Figure 16: Διάλογος Αξιολόγησης



Figure 17: Αποτροπή Ψηφοφορίας για δεύτερη φορά

διαδικασία ολοκληρωθεί επιτυχημένα και ο Server αποστέλλει μήνυμα επιτυχούς εισαγωγής τότε το Activity στην αρχική κατάσταση.

## 2. Αξιολόγηση της εγκυρότητας των κοντινών του σημείων στάθμευσης που έχουν εισαχθεί στο σύστημα από άλλους χρήστες.

Μόλις ο χρήστης πατήσει το αντίστοιχο κουμπί της διεπαφής τότε εκτελείται ασύγχρονα http Post request προς το Server με παραμέτρους την τρέχουσα εκτίμηση τοποθεσίας ως γεωγραφικό μήκος και πλάτος (Latitude, Longitude).

Εφόσον υπάρχουν σε ακτίνα 20 μέτρων από την τρέχουσα τοποθεσία του χρήστη σημεία στάθμευσης, και αυτά δεν έχουν ήδη αξιολογηθεί από τον ίδιο χρήστη στο παρελθόν τότε ο χρήστης καλείται να αξιολογήσει την εγκυρότητα τους. Επιλέγοντας σημείο προς αξιολόγηση εμφανίζεται ένα παράθυρο διαλόγου με το οποίο ο χρήστης ψηφίζει θετικά ή αρνητικά. Τέλος γίνεται ασύγχρονα και πάλι http Post request προς τον Server με παραμέτρους το σημείο που αξιολογήθηκε και το αποτέλεσμα της ψηφοφορίας. Ο Server θα ενημερώσει τον χρήστη για την επιτυχία ή όχι του αιτήματος ψηφοφορίας και το αντίστοιχο σημείο θα αποθηκευτεί στην τοπική βάση δεδομένων του Android (Sql Lite) έτσι ώστε να αποτρέψει τον χρήστη να ψηφίζει περισσότερες από μια φορές το ίδιο σημείο.

### 3.2 Driver Activity

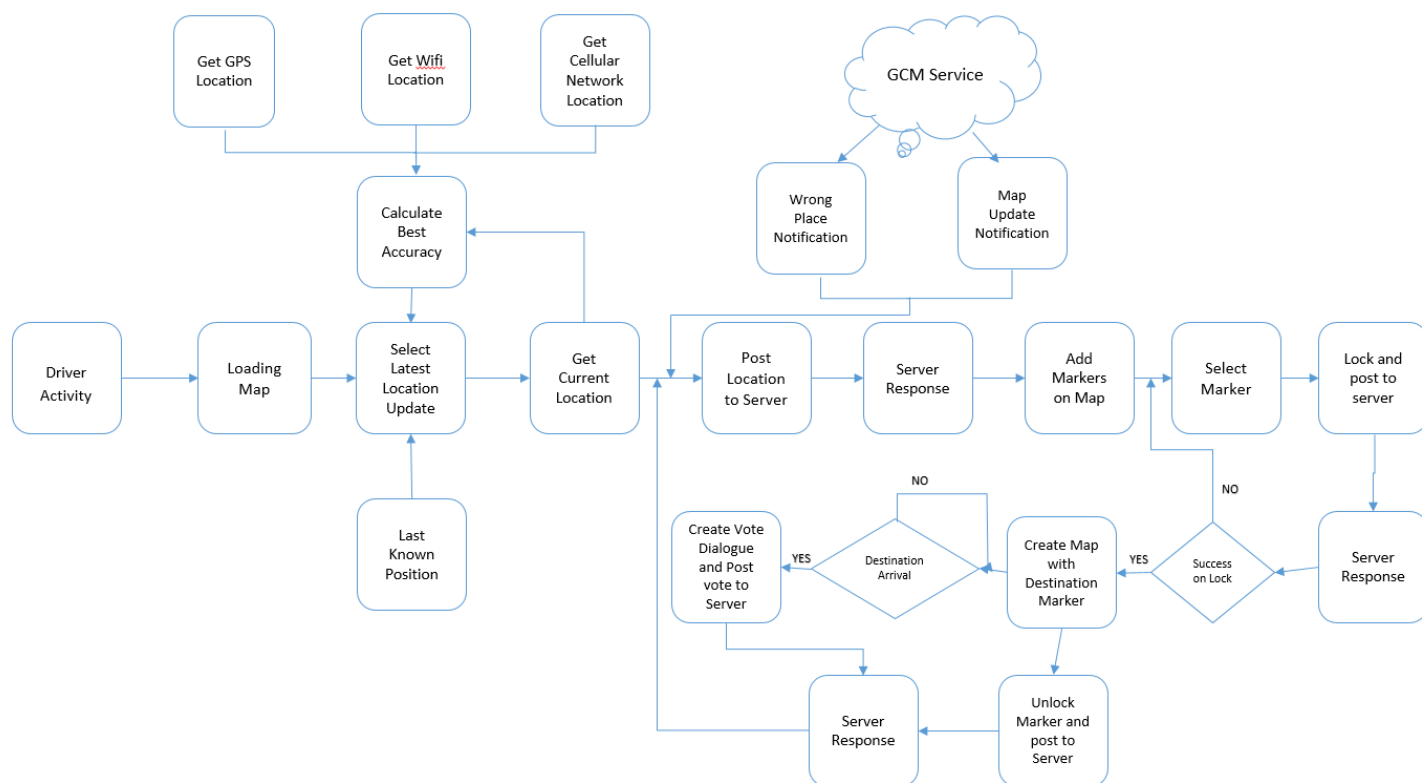


Figure 18: Διάγραμμα Driver Activity

Με την εκκίνηση της λειτουργία Driver που είναι παρόμοια με τη λειτουργία Pedestrian η εφαρμογή αρχικοποιεί το Map Fragment για την εμφάνιση του χάρτη και ανακτά την τοποθεσία με τον ίδιο τρόπο που περιγράφηκε στην λειτουργία Pedestrian. Αφού ανακτηθεί η τοποθεσία εκτελείται ασύγχρονα Http Post request προς τον Server με παραμέτρους τις γεωγραφικές συντεταγμένες του χρήστη. Στη συνέχεια εκτελείται η βασική λειτουργικότητα της εφαρμογής μας που είναι η τοποθέτηση πάνω στον χάρτη με μορφή Markers των δέκα κοντινότερων θέσεων στάθμευσης.



Figure 19: Επιλογή σημείου προς δέσμευση πάνω στον χάρτη

```
{
  "places": [
    {
      "lng": "22.946146666666664",
      "Conf": "0.016231752262825985",
      "Category": "green",
      "lat": "39.360796666666667",
      "name": "Dimitriados 137-141"
    },
    {
      "lng": "22.946061666666665",
      "Conf": "0.09479858757337126",
      "Category": "red",
      "lat": "39.359813333333335",
      "name": "Άμ009 A.Kartali 13"
    },
    {
      "lng": "22.944816666666664",
      "Conf": "0.09479858757337126",
      "Category": "orange",
      "lat": "39.361443333333334",
      "name": "Δημητριάδος 44-50"
    }
  ]
}
```

Στη συνέχεια εφόσον ο οδηγός ενδιαφέρεται για κάποια από τις θέσεις στάθμευσης που έχει λάβει, την επιλέγει πατώντας πάνω στον αντίστοιχο Marker. Τότε του δίνεται η δυνατότητα να 'κλειδώσει' την συγκεκριμένη θέση πατώντας το σχετικό κουμπί που παρέχει η διεπαφή χρήστη και γίνεται επικοινωνία με τον Server. Η συγκεκριμένη κίνηση οδηγεί σε δύο πιθανά σενάρια εκτέλεσης:

1. Η θέση δεσμεύεται επιτυχημένα με αποτέλεσμα να μην είναι διαθέσιμη σε γειτονικούς χρήστες που την έχουν λάβει σαν κοντινή θέση στάθμευσης και να μην είναι ορατή σε κανέναν άλλον οδηγό.

- Υπάρχει αποτυχία στη δέσμευση της συγκεκριμένης θέσης για τους εξής δυο λόγους. Είτε επειδή το σημείο έχει δεσμευτεί από άλλον κοντινό χρήστη, είτε επειδή το σημείο δεν είναι πλέον διαθέσιμο στη βάση δεδομένων. Σε αυτή την περίπτωση εμφανίζονται τα ανανεωμένα κοντινότερα σημεία και καλείται να επιλέξει εκ νέου.

Όταν η δέσμευση γίνει με επιτυχία τότε ο χάρτης ανανεώνεται και εμφανίζει μόνο το σημείο που έχει δεσμευτεί δίνοντας δυνατότητα στον χρήστη είτε να αποδεσμεύσει το δεσμευμένο σημείο είτε να συνεχίσει την πορεία του προς αυτό το σημείο ακολουθώντας την διαδρομή.

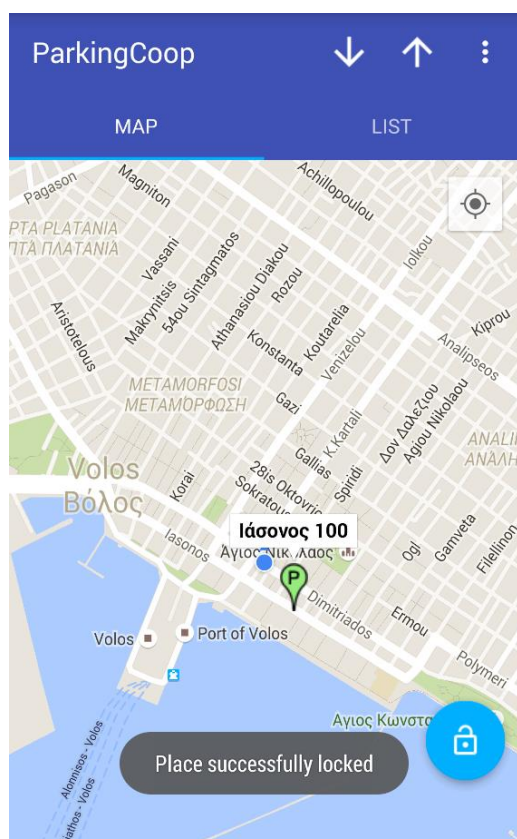


Figure 20:Επιτυχής δέσμευση Θέσης Στάθμευσης

Στην περίπτωση που αποδεσμεύει την θέση γίνεται ενημέρωση προς το Server ότι η θέση είναι ξανά διαθέσιμη προς όλους τους χρήστες και η εφαρμογή επιστρέφει στην εμφάνιση του χάρτη με τα ανανεωμένα δέκα κοντινότερα σημεία. Στη διαφορετική περίπτωση που ο χρήστης επιμένει και ακολουθεί την διαδρομή προς το δεσμευμένο σημείο, τότε μόλις η απόσταση του χρήστη και του σημείου γίνει μικρότερη από δέκα μέτρα, η εφαρμογή υποθέτει ότι ο οδηγός έχει φτάσει στον προορισμό του και εμφανίζεται ένα παράθυρο διαλόγου μέσω του οποίου γίνεται αξιολόγηση της εγκυρότητας της πληροφορίας και η ψηφοφορία στέλνεται στον Server. Τέλος τερματίζεται η συγκεκριμένη δραστηριότητα (Driver Activity) και ο χρήστης επιστρέφει στο σημείο επιλογής κατάστασης (Driver, Pedestrian).

### **3.3 Ειδικές Περιπτώσεις**

Η ροή εκτέλεσης που περιγράφηκε παραπάνω μπορεί να μεταβληθεί λόγω λήψεως ειδοποιήσεων. Ο χρήστης μπορεί να λάβει ειδοποιήσεις από την υπηρεσία Google Cloud Messaging στις εξής δύο περιπτώσεις:

- Όταν προστίθεται σημείο το οποίο είναι σε πιο κοντινή απόσταση από το πιο απομακρυσμένο διαθέσιμο προς αυτόν σημείο. Σε αυτήν την περίπτωση η ειδοποίηση

ενημερώνει τον χρήστη ότι έχει προστεθεί νέο σημείο σε κοντινή απόσταση και του δίνει τη δυνατότητα να ανανεώσει τον χάρτη του μόλις πατήσει πάνω στην διεπαφή της ειδοποίησης.

2. Όταν ένας οδηγός έχει δεσμεύσει μια θέση στάθμευσης και προτού φτάσει στον προορισμό του η θέση κρίνεται άκυρη από το σύστημα, τότε λαμβάνει την αντίστοιχη ειδοποίηση, ακυρώνεται η δέσμευση και ανανεώνεται ο χάρτης του.



## 4.ΛΕΙΤΟΥΡΓΙΕΣ SERVER

Ο Server της εφαρμογής εκτελεί τις εξείς λειτουργίες μέσω των οποίων οργανώνει όσο γίνεται πιο αποδοτικά την διαδικασία εύρεσης στάθμευσης.

1. Login (User Authentication)
2. Διαχείριση Driver
3. Διαχείριση Pedestrian
4. Διαχείριση Βάσης Δεδομένων
  1. Δέσμευση Θέσης
  2. Αποδέσμευση θέσης
  3. Ψηφοφορία
5. Οργάνωση ειδοποιήσεων
6. Αλγόριθμος Αξιολόγησης της αξιοπιστίας των χρηστών
7. Χρονολόγηση θέσεων στάθμευσης

### 4.1 Πίνακες SQL

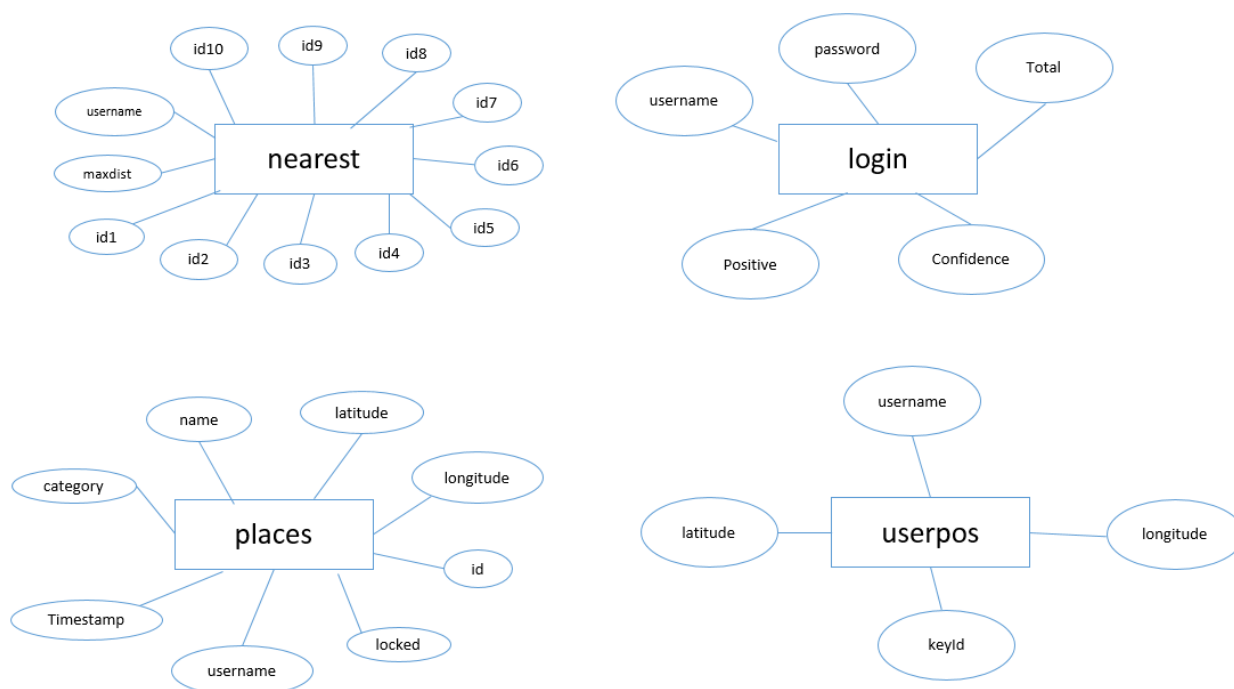


Figure 21: Διάγραμμα Βάσης Δεδομένων

Στη βάση δεδομένων αποθηκεύονται όλες οι απαραίτητες πληροφορίες για την λειτουργία της εφαρμογής. Αυτό γίνεται με την χρήση τεσσάρων πινάκων όπως φαίνεται στο παραπάνω διάγραμμα.

## **4.2 User Authentication**

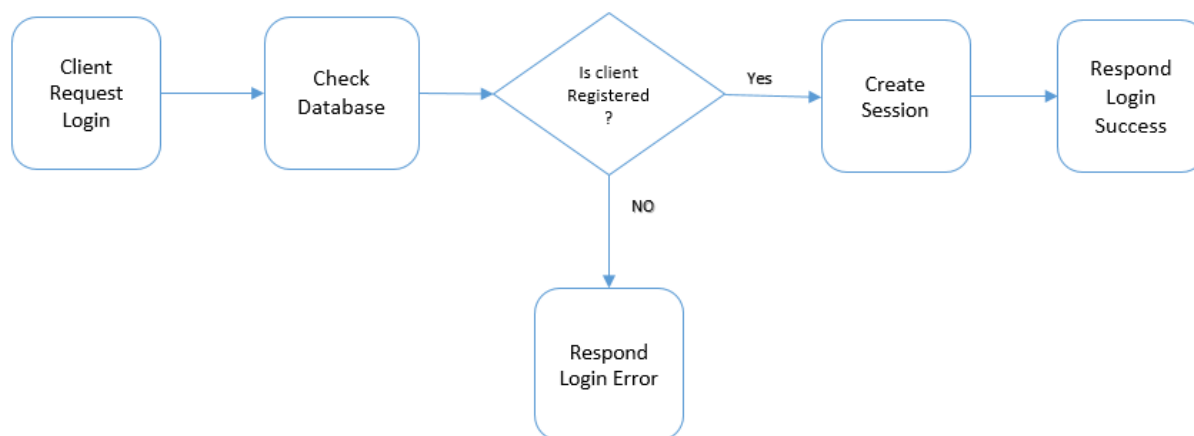


Figure 22: Διάγραμμα ροής για ταυτοποίηση χρήστη

Ο Server δέχεται ένα http αίτημα με παραμέτρους username, password και ελέγχει αν τα στοιχεία εισόδου έχουν καταχωρηθεί στη βάση δεδομένων. Αν ναι, τότε ο Server ταυτοποιεί τον χρήστη επιτυχημένα και του επιτρέπει να κάνει χρήση της εφαρμογής. Διαφορετικά η ταυτοποίηση αποτυγχάνει και ο Server μέσω http ειδοποιεί τον χρήστη με το κατάλληλο μήνυμα σφάλματος.

## **4.3 Διαχείριση Driver Mode**



Figure 23: Διάγραμμα Διαχείρισης Driver



Ο Οδηγός μέσω http αίτησης προς το Server στέλνει τις συντεταγμένες του και το μοναδικό κλειδί επικοινωνίας του που έχει λάβει από την υπηρεσία Google Cloud Messaging. Ο Server καταχωρεί στους κατάλληλους πίνακες τις πληροφορίες και υπολογίζει και καταχωρεί στον σχετικό πίνακα τις δέκα κοντινότερες θέσεις στάθμευσης. Τέλος δημιουργείται και αποστέλλεται ένα αντικείμενο που ακολουθεί την δομή Json και περιέχει όλες τις απαραίτητες πληροφορίες.

#### **4.4 Διαχείριση Pedestrian**

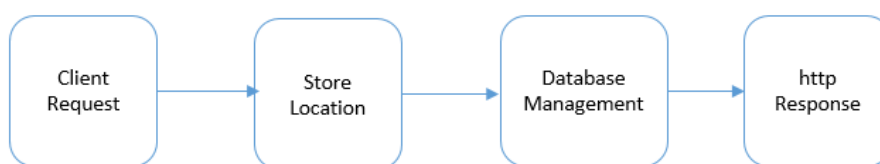
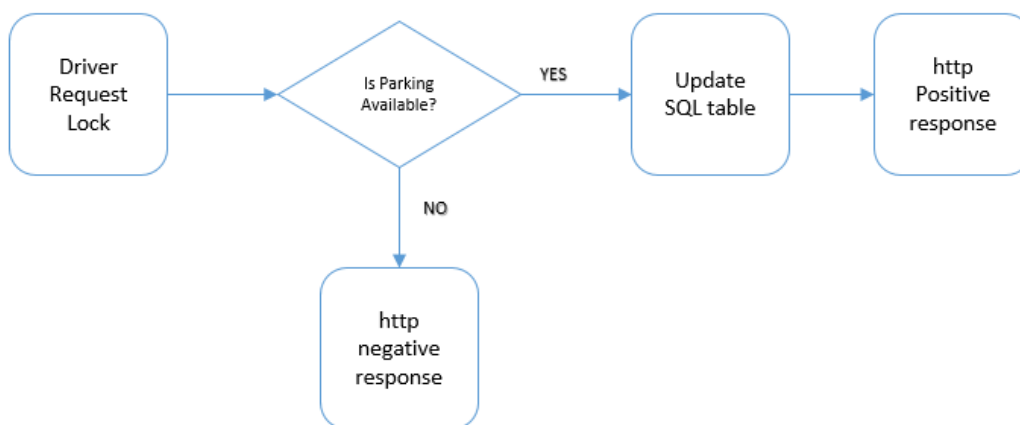


Figure 24: διάγραμμα διαχείρισης Pedestrian

Ο πεζός μέσω http αίτησης προς το Server στέλνει την τοποθεσία του η οποία αποθηκεύεται στον κατάλληλο πίνακα. Στη συνέχεια μπορεί να εκτελέσει λειτουργίες που απαιτούν διαχείριση βάσης δεδομένων. Ο server απαντάει μέσω http για το αποτέλεσμα της αίτησης.

#### **4.5 Διαχείριση Βάσης Δεδομένων**

##### **4.5.1 Δέσμευση ή αποδέσμευση Θέσης**



Εφόσον ένας οδηγός αιτηθεί να δεσμεύσει ή να αποδεσμεύσει μια θέση στάθμευσης ο Server εκτελεί την εξής διαδικασία. Λαμβάνεται ως παράμετρος από την http αίτηση το μέρος που ο χρήστης επιθυμεί να δεσμεύσει ή να αποδεσμεύσει και στη συνέχεια γίνεται έλεγχος στον πίνακα places και εφόσον το σημείο είναι διαθέσιμο ανανεώνεται το πεδίο locked του πίνακα και αποστέλλεται http απάντηση για την επιτυχία ή την αποτυχία της αίτησης.

#### 4.5.2 Ψηφοφορία

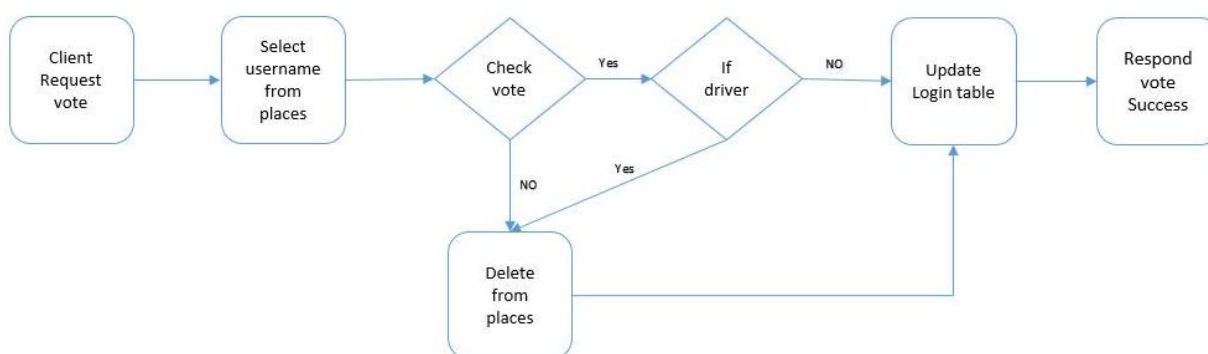


Figure 25: Διάγραμμα ροής για αξιολόγηση

Σε περίπτωση που ένα χρήστης επιθυμεί να αξιολογήσει την εγκυρότητα μιας θέσης στάθμευσης ο Server εκτελεί την ακόλουθη διαδικασία. Μέσω της http αίτησης που δέχεται λαμβάνει παραμέτρους το username του χρήστη που επιθυμεί να ψηφίσει και το μέρος στο οποίο θα εφαρμοστεί η ψηφοφορία. Στη συνέχεια γίνεται αναζήτηση του σημείου στον πίνακα places και λαμβάνεται το username

του χρήστη που έχει δημοσιεύσει την πληροφορία. Αν η ψήφος είναι θετική τότε ανανεώνονται τα πεδία Total, Positive που αφορούν τον χρήστη που δημοσίευσε το σημείο στον πίνακα login. Διαφορετικά δεν ανανεώνεται το πεδίο Positive και το σημείο διαγράφεται από τον πίνακα places. Σε περίπτωση που ο χρήστης που αιτείται ψηφοφορία είναι 'Driver' τότε το μέρος που αξιολογεί θα διαγραφεί από τον πίνακα places σε περίπτωση θετικής ψήφου καθώς θεωρούμε ότι πάκκαρε επιτυχημένα.

## **4.6 Οργάνωση ειδοποιήσεων**

Ο Server στέλνει στους χρήστες δύο είδη ειδοποιήσεων στις εξής περιπτώσεις

1. Προσθήκη νέου σημείου
2. Αρνητική Ψηφοφορία σε μια υπάρχουσα θέση στάθμευσης

### **4.6.1 Προσθήκη νέου σημείου**

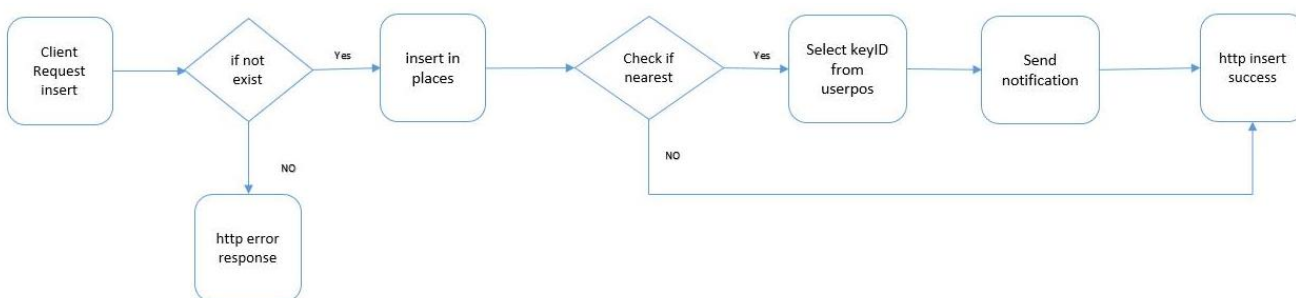


Figure 26: Διάγραμμα ροής ειδοποίησης σε περίπτωση εισαγωγής νέας θέσης

Σε περίπτωση που ένα νέο διαθέσιμο σημείο στάθμευσης εισαχθεί στη βάση δεδομένων τότε γίνεται έλεγχος σε όλους τους οδηγούς για το αν το νέο σημείο πρέπει να προστεθεί στη λίστα των δέκα κοντινότερων. Εφόσον η παραπάνω συνθήκη είναι αληθής τότε στέλνεται ειδοποίηση σε όλους χρήστες κρίνεται απαραίτητο χρησιμοποιώντας την υπηρεσία GCM Cloud Messaging δίνοντας σαν παράμετρο τον μοναδικό κωδικό επικοινωνίας που ο οδηγός έχει καταχωρήσει στο σύστημα.

#### 4.6.2 Αρνητική Ψηφοφορία σε μια υπάρχουσα θέση στάθμευσης

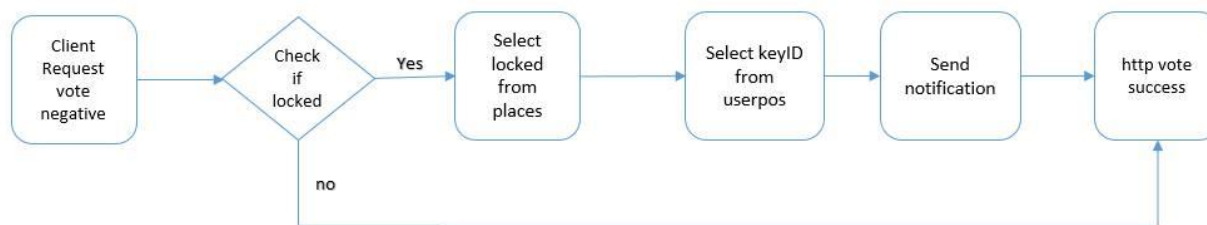


Figure 27: Διάγραμμα ροής για ειδοποίηση σε περίπτωση αρνητικής αξιολόγησης ενός σημείου

Σε περίπτωση που ένας πεζός ψηφίσει αρνητικά για μια συγκεκριμένη θέση στάθμευσης και τη συγκεκριμένη στιγμή κάποιος οδηγός την έχει ήδη δεσμεύσει τότε για να αποφευχθεί επιπλέον κόπος από πλευράς του λαμβάνει την αντίστοιχη ειδοποίηση, η οποία τον προειδοποιεί ότι το μέρος που έχει δεσμεύσει είναι άκυρο.

#### 4.7 Αξιολόγηση της αξιοπιστίας των χρηστών μέσω της Μεθόδου Wilson

Η αξιοπιστία των δεδομένων σε μια εφαρμογή που βασίζεται στις πληροφορίες των χρηστών της είναι ζωτικής σημασίας. Συνεπώς η αξιολόγηση των χρηστών οι οποίοι είναι και μοναδικοί πάροχοι πληροφορίας πρέπει να γίνεται με αποδοτικό και όσο το δυνατόν δίκαιο τρόπο. Σκοπός της εφαρμογής είναι να υλοποιήσει ένα δείκτη αξιοπιστίας των χρηστών (Confidence) και κατά συνέπεια των δεδομένων που αυτοί παρέχουν. Ο δείκτης αξιοπιστίας των χρηστών της εφαρμογής (Confidence rate) στηρίζεται στον υπολογισμό του κάτω φράγματος της μεθόδου αξιοπιστίας Wilson για παραμέτρους τύπου Bernoulli.

$$\left( \hat{p} + \frac{z_{\alpha/2}^2}{2n} \pm z_{\alpha/2} \sqrt{[\hat{p}(1 - \hat{p}) + z_{\alpha/2}^2/4n]/n} \right) / (1 + z_{\alpha/2}^2/n).$$

Figure 28: Wilson Confidence Interval

n=Μέγεθος δείγματος

p=(positive/n)

a=Επιθυμητή αξιοπιστία

z(a/2)=τιμή του επιπέδου της επιθυμητής αξιοπιστίας. Στην εφαρμογή μας η επιθυμητή αξιοπιστία είναι 95% συνεπώς Z(a/2)=1.96.

Μια άλλη επιλογή θα ήταν να χρησιμοποιήσουμε τον Στατιστικό μέσο όρο (positive/total).

Αυτό θα είχε σαν αποτέλεσμα σε μικρό αριθμό δείγματος να βγάζει αναξιόπιστο συμπέρασμα.

Για παράδειγμα ένας χρήστης με (5/5) positive έχει confidence 100% ενώ ένας άλλος με (80/100) έχει confidence 80% ενώ στην πραγματικότητα έχουμε περισσότερα στατιστικά για αυτόν τον χρήστη ο οποίος είναι μακροπρόθεσμα πιο αξιόπιστος από τον πρώτο. Το παραπάνω πρόβλημα λύνεται με την χρήση της μεθόδου Wilson γιατί δίνει βάρος στην μακροπρόθεσμη αξιοπιστία του χρήστη. Όσο μεγαλώνει το δείγμα τόσο αυξάνεται η αξιοπιστία του συμπεράσματος χωρίς να έχουμε ακραίες τιμές σε μικρά δείγματα. Μακροπρόθεσμα οι 2 μέθοδοι συγκλίνουν.

Στο σχήμα (Figure 29) βλέπουμε την διακύμανση της τιμής Wilson ενός χρήστη ο οποίος δίνει

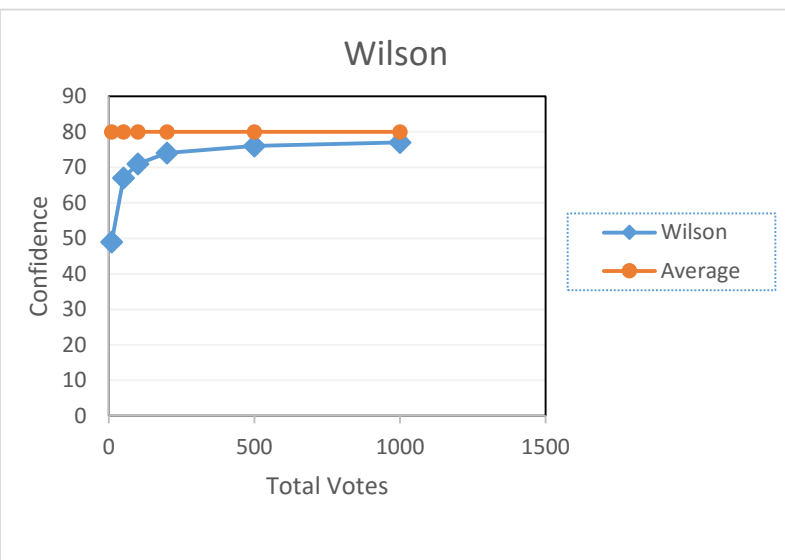


Figure 29: Αξιοπιστία χρήστη με ποσοστό εγκυρότητας 80%

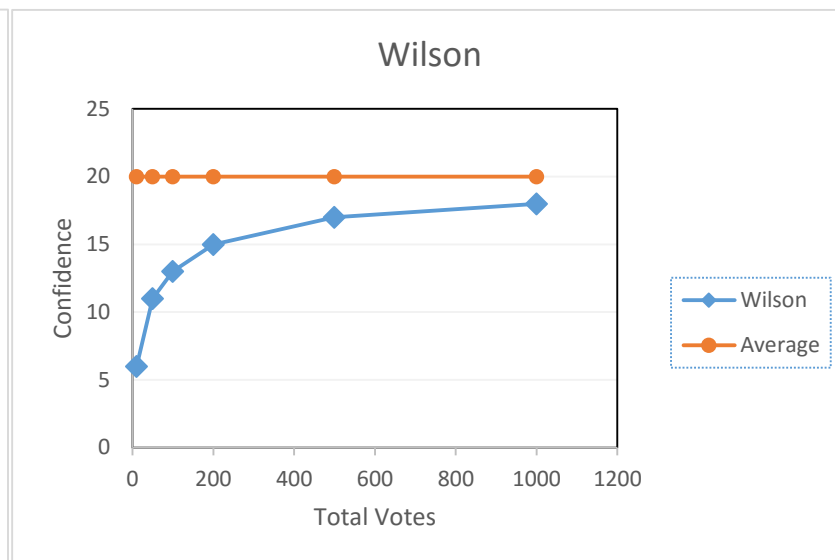


Figure 30: Αξιοπιστία χρήστη με ποσοστό εγκυρότητας 20%

κατά 80% έγκυρες πληροφορίες. Παρατηρούμε ότι όσο πιο ενεργός είναι ένας χρήστης τόσο το σύστημα τον θεωρεί πιο αξιόπιστο και η τιμή Wilson συγκλίνει στον μέσο όρο. Στο σχήμα (Figure 30) διακρίνεται η συμπεριφορά με ποσοστό εγκυρότητας 20%.

Παρατηρούμε ότι η αξιοπιστία του συγκεκριμένου χρήστη συγκλίνει πιο αργά στο μέσο όρο. Στην περίπτωση της εφαρμογής μας κάθε φορά που κάποιος χρήστης αξιολογεί την εγκυρότητα ενός σημείου εκτελείται η μέθοδος Wilson πάνω στα δεδομένα Positive και Total του χρήστη του οποίου η πληροφορία αξιολογήθηκε. Το αποτέλεσμα της μεθόδου εισάγεται στο πεδίο Confidence και αποτελεί τον δείκτη αξιοπιστίας των χρηστών του συστήματος.

#### **4.8 Χρονολόγηση σημείων**

Πολύ σημαντικό στην αξιοπιστία της εφαρμογής μας, είναι ο χρόνος που μεσολαβεί από την στιγμή που δημοσιεύεται μια διαθέσιμη θέση στάθμευσης μέχρι τη στιγμή που κάποιος την δεσμεύσει. Για παράδειγμα μια θέση η οποία δεσμεύεται τρία λεπτά αφότου δημοσιεύεται έχει περισσότερες πιθανότητες να είναι ακόμα διαθέσιμη τη στιγμή που στιγμή που φτάσει εκεί ο οδηγός απ ότι αν είχε δημοσιευτεί πενήντα λεπτά πριν.

Για αυτό το λόγο εισάγεται στο σύστημα μια επιπλέον πληροφορία σχετικά με το πόση ώρα σε λεπτά έχει περάσει από την στιγμή της δημοσίευσης μέχρι την τρέχουσα χρονική στιγμή. Για να είναι πιο φιλική προς το χρήστη αυτή η πληροφορία, κατηγοριοποιείται σε 4 κατηγορίες.

- *Green*: Σημεία που δημοσιεύτηκαν τα προηγούμενα 0-5 λεπτά
- *Yellow*: Σημεία που δημοσιεύτηκαν τα προηγούμενα 5-15 λεπτά
- *Orange*: Σημεία που δημοσιεύτηκαν τα προηγούμενα 15-30 λεπτά
- *Red*: Σημεία που δημοσιεύτηκαν τα προηγούμενα 30-60 λεπτά

Οποιοδήποτε σημείο δεν ανήκει στις παραπάνω κατηγορίες θεωρείται από το σύστημα άκυρο και διαγράφεται καθώς είναι πολύ πιθανό να μην είναι πλέον η πληροφορία έγκυρη.

## 5. Evaluation

Η απόδοση του συστήματος που περιγράφηκε πρέπει να είναι όσο το δυνατόν πιο κοντά στις απαιτήσεις του μέσου χρήστη έχοντας σαν στόχο μια όσο το δυνατόν πιο ικανοποιητική εμπειρία χρήσης (User Experience). Οι παράγοντες που καθορίζουν σημαντικά την απόδοση της εφαρμογής είναι τόσο η βέλτιστη διαχείριση των πόρων (μνήμη, CPU) όσο και οι ικανοποιητικοί χρόνοι απόκρισης όλων των διαθέσιμων ενεργειών.

### 5.1 Μεθοδολογία

Βασικός και κρίσιμος παράγοντας για την βέλτιστη απόδοση της εφαρμογής μας είναι οι ικανοποιητικοί χρόνοι απόκρισης τόσο από την σκοπιά ενός οδηγού που αναζητά θέση στάθμευσης όσο και ενός πεζού που είναι πρόθυμος να παρέχει έγκυρες πληροφορίες στο σύστημα.

Ως σύστημα το οποίο θα κριθεί σε δοκιμασία (System Under Test) ορίζουμε την κινητή συσκευή που εκτελεί την εφαρμογή μαζί με τον server που παρέχει τις πληροφορίες αλλά και όλες τις εμπλεκόμενες cloud υπηρεσίες που περιγράφηκαν παραπάνω. Η βασική μετρική που εισάγουμε για την αξιολόγηση της ποιότητας των υπηρεσιών που παρέχει, είναι ο χρόνος απόκτησης (Response Time) από την στιγμή που ο χρήστης θα κάνει μια αίτηση προς το σύστημα μέχρι να λάβει απάντηση. Παράμετρος αξιολόγησης του χρόνου απόκρισης είναι η κινητή συσκευή Samsung Galaxy S4 με την οποία εκτελέστηκαν τα πειράματα.

Βασικός παράγοντας που εξετάζεται και καθορίζει τον χρόνο απόκρισης είναι η ποιότητα του δικτύου, αλλά και το πρωτόκολλο ασύρματης δικτύωσης που έχει επιλεγεί την εκάστοτε στιγμή για την μετάδοση της πληροφορίας.

<i>Parameters</i>			<i>Factors</i>	
<i>Mobile</i>		<i>Environment</i>		
<i>CPU Type</i>	<i>Qualcomm MSM8974</i>	<i>Web Server: Apache Tomcat v7</i>	<i>Network</i>	<i>3G</i>
<i>Screen Size</i>	<i>5"</i>			<i>2G</i>
<i>RAM</i>	<i>2 GB</i>			
<i>OS</i>	<i>Android 4.4.2</i>			

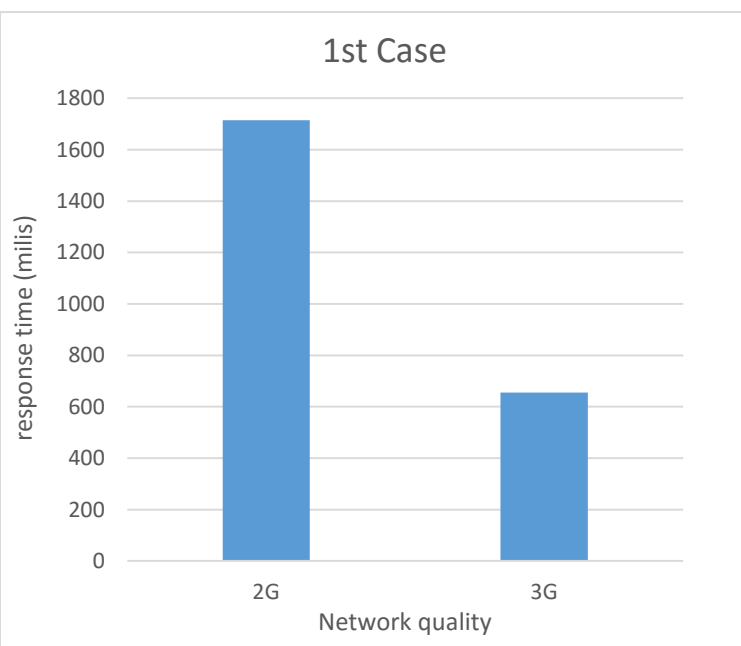
*Table 1 Parameters and Factors Defined for Performance Evaluation*



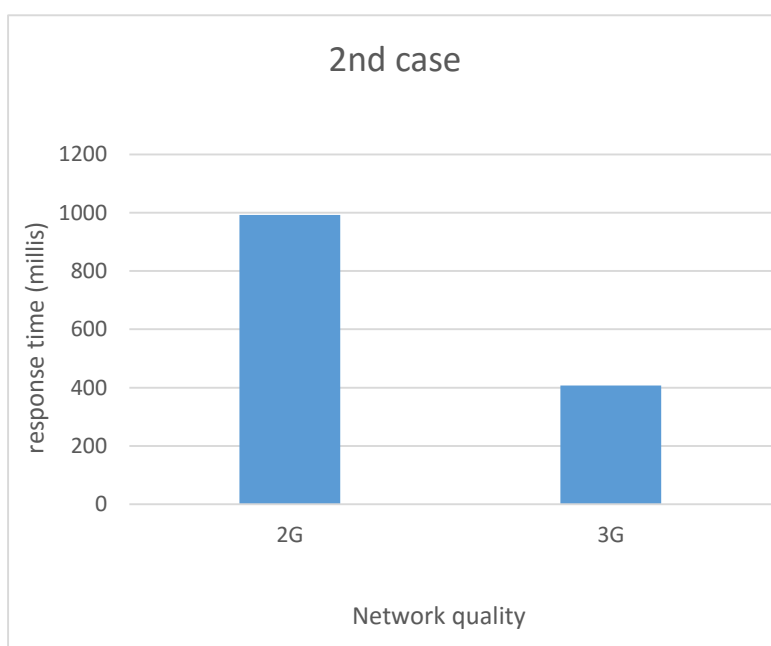
## **5.2 Πειράματα**

Τα βασικά Σενάρια ως προς τα οποία αξιολογήσαμε το σύστημα με βάση τον χρόνο απόκρισης συναρτήσει των παραπάνω παραμέτρων είναι τα εξής.

1. Ο Οδηγός αιτείται τα κοντινότερα Parking
2. Λειτουργίες αλληλεπίδρασης με τη βάση δεδομένων
3. Ανάκληση πληροφορίας από την υπηρεσία Geocoding της Google
4. Λήψη ειδοποιήσεων από Google Cloud.



*Figure 32:Response Time in 1st Case*



*Figure 31:Response Time in 2nd Case*

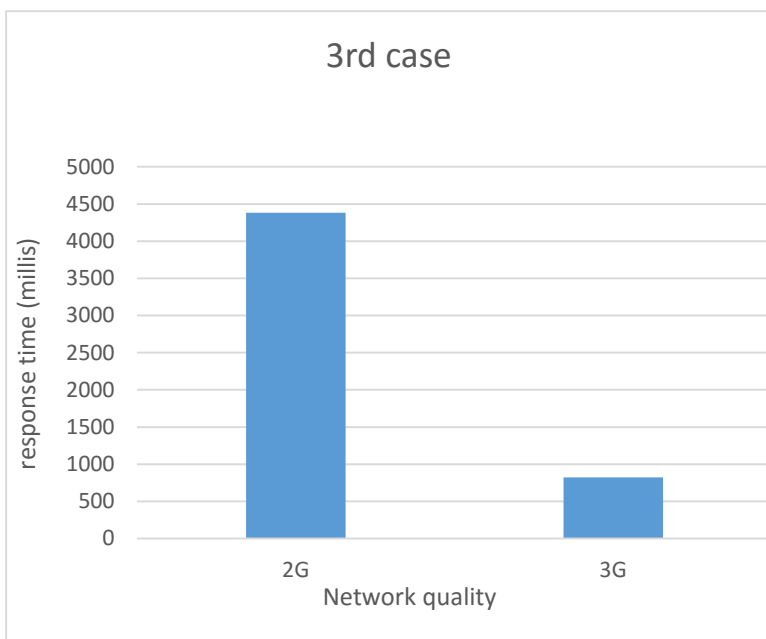


Figure 33: Response Time in 3rd Case

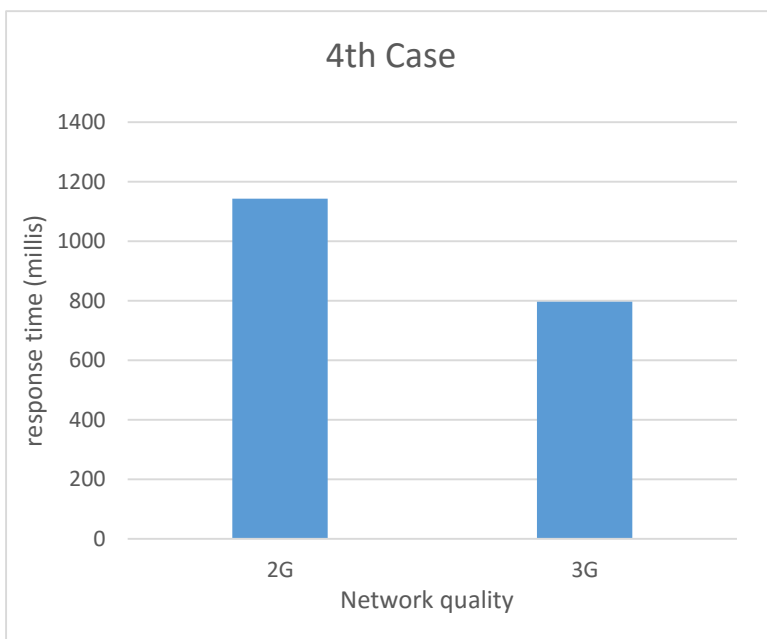


Figure 34: Response Time in 4th Case

Τα πειράματα εκτελέστηκαν σε αστικό περιβάλλον υπό κάλυψη διαφορετικής ποιότητας δικτύου. Έγιναν πενήντα καταμετρήσεις σε καθένα από τα τέσσερα σενάρια και οι μέσοι όροι απόκρισης φαίνονται στα παραπάνω διαγράμματα. Στην περιοχή που εκτελέστηκαν τα πειράματα δεν υπήρχε δυνατότητα για 4G δικτύου. Σε περίπτωση που κάτι τέτοιο ήταν δυνατό σε όλα τα σενάρια οι χρόνοι απόκρισης θα ήταν σημαντικά μειωμένοι. Για την επιτυχία του εγχειρήματος που παρουσιάσαμε στα πλαίσια αυτής της εργασίας είναι σημαντικά τόσο η καθολική συμμετοχή των πολιτών όσο και η άρτια ποιότητα υπηρεσιών δικτύωσης που μπορεί να παρέχει ένα αστικό περιβάλλον.

## 6. Συμπεράσματα-Επεκτάσεις

### 6.1 Συμπεράσματα

Το πρόβλημα στάθμευσης σε περιβάλλον Smart City αποτελεί ένα κύριο πρόβλημα προς επίλυση. Η προσέγγιση μας στηρίζεται σε μια κοινωνική και συνεργατική συμπεριφορά των σύγχρονων πολιτών των οποίων η μαζική συμμετοχή αποτελεί βασική προϋπόθεση για την επιτυχία του εγχειρήματος. Όσο αυξάνεται ο αριθμός και η συμμετοχή των χρηστών τόσο βελτιώνεται η ποιότητα των υπηρεσιών και η εγκυρότητα του δείκτη αξιοπιστίας.

### 6.2 Επεκτάσεις

Η εφαρμογή μας καλύπτει σε ικανοποιητικό βαθμό όλες τις περιπτώσεις και είναι σε θέση να συμβάλει σημαντικά στην επίλυση του προβλήματος της στάθμευσης. Υπάρχει δυνατότητα περαιτέρω βελτίωσης τόσο στην παροχή κινήτρων προς τους πολίτες ώστε να είναι ολοένα και πιο ενεργοί ως χρήστες, όσο και στον εντοπισμό και εν τέλει στην απομόνωση των κακόβουλων χρηστών, δηλαδή χρηστών οι οποίοι δημοσιεύουν σκόπιμα ψευδή δεδομένα ή αξιολογούν ψευδώς άλλους χρήστες.

## Βιβλιογραφία

- [1].Last Na Annalisa Cocchia, “Smart and Digital City: A Systematic Literature Review”, Part of the series Progress in IS pp 13-43, June 2014
- [2].M.Y.I.Idris, Y.Y. Leng, E.M. Tamil, N.M. Noor and Z.Razak, “Car Park System: A Review of Smart Parking System and its Technology”, Information Technology Journal 8 (2) pp 101-113, 2009
- [3].Amin Kianpisheh, Norlia Mustaffa, Pakapan Limtrairut and Pantea Keikhosrokiani, “Smart Parking System (SPS) Architecture Using Ultrasonic Detector”, International Journal of Software Engineering and Its Applications, Vol.6 Issu3, p51, July 2012
- [4].Feng Xia, Laurence T. Yang, Lizhe Wang and Alexey Vinel “Internet of Things”, International Journal of Communication Systems, Volume 25, Issue 9, pages 1101–1102, September 2012
- [5].Vermesan O., Friess P., Guillemin P., Gusmeroli S., Sundmaeker H., Bassi A., Jubert I.S., Mazura M., Harrison M., Eisenhauer M. “Internet of things strategic research roadmap”. *Int. Things-Global Technol. Soc. Trends* 2011, 1, 9–52
- [6].Boussier J.M., Estraillier P., Sarramia D., Augeraud M. Using agent-based of driver behavior in the context of car park optimization. Proceedings of the 3rd International IEEE Conference on Intelligent Systems, London, UK, 4–6 September 2006; pp. 395–400
- [7].Zhanlin Ji , Ivan Ganchev , Máirtín O'Droma , Li Zhao and Xueji Zhang, “A Cloud-Based Car Parking Middleware for IoT-Based Smart Cities: Design and Implementation”, *Sensors* 2014, 14(12), 22372-22393; doi:10.3390/s141222372
- [8].Meier R. “*Professional Android 4 Application Development*”, John Wiley & Sons: New York, NY, USA, 2012
- [9].Lawrence D. Brown, T. Tony Cai and Anirban DasGupta, “Interval Estimation for a Binomial Proportion”, *Statistical Science*, Vol. 16, No. 2 (May, 2001), pp. 101-117
- [10]. Thiago Soares Fernandes, Erika Cota and Alvaro Freitas Moreira, “Performance Evaluation of Android Applications: a Case Study”, 2014 Brazilian Symposium on Computing Systems Engineering, pp 79-84, November 2014